SCHOOL OF
CIVIL ENGINEERING

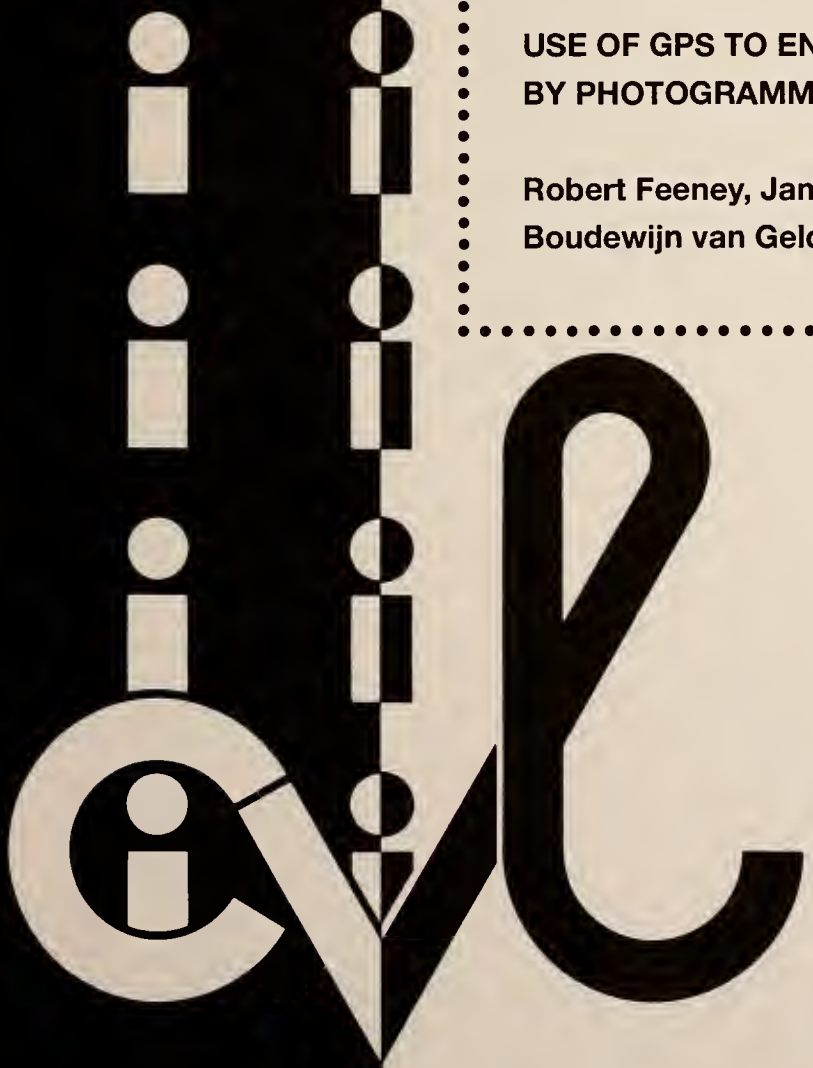
INDIANA
DEPARTMENT OF TRANSPORTATION

JOINT HIGHWAY RESEARCH PROJECT

FHWA/IN/JHRP-95/6
Final Report

USE OF GPS TO ENHANCE MAPPING BY PHOTOGRAMMETRY

Robert Feeney, James Bethel,
Boudewijn van Gelder, and Steve Johnson



PURDUE UNIVERSITY

Final Report on

USE OF GPS TO ENHANCE MAPPING BY PHOTOGRAMMETRY

by

Robert Feeney, James Bethel, Boudewijn van Gelder, and Steve Johnson

Project No. C-36-72B

File No. 1-6-2

HPR Study in cooperation with

Indiana Department of Transportation

and

Federal Highway Administration

Purdue University
School of Civil Engineering
West Lafayette, IN 47907

May 14, 1996

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/IN/JHRP-95/6		

4. Title and Subtitle	5. Report Date
USE OF GPS TO ENHANCE MAPPING BY PHOTOGRAMMETRY	May 14, 1996
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Robert Feeney, James Bethel, Boudewijn van Gelder, Steve Johnson	FHWA/IN/JHRP-95/6

9. Performing Organization Name and Address	10. Work Unit No.
Joint Highway Research Project Civil Engineering Building Purdue University West Lafayette, Indiana 47907-1284	11. Contract or Grant No.

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Indiana Department of Transportation State Office Building 1000 North Senate Avenue Indianapolis, IN 46204	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the Indiana Department of Highways and Federal Highway Administration.

16. Abstract

This research has sought to demonstrate the benefits of kinematic GPS in conjunction with photogrammetric aerial triangulation for mapping purposes. The work was performed with the existing INDOT aerial camera which is quite old, and while this affected the accuracy of the results, the demonstration of capability with a likely potential for excellent results was achieved. Ground point accuracies from the strip were in the sub-meter range with only a single control point. This is contrasted with full-model control requiring 50-80 control points for such a strip. In order to bring accuracies down to an acceptable level, i.e. sub decimeter, it will be necessary to retrofit the INDOT camera with a better shutter event signal (or obtain a new camera). A computer program for least squares adjustment of independent models has been developed. This would permit processing of data from INDOT Wild B8 stereoplotters. With a few implementation steps, INDOT would have an operational and production capability yielding significant productivity improvements.

17. Key Words	18. Distribution Statement
GPS, photogrammetry, aerial triangulation, camera calibration, mapping	No restrictions. This document is available to the public through the National Technical Information Service, Virginia, 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	89	

Form DOT F 1700.7 (8-69)

IMPLEMENTATION REPORT

This research has sought to demonstrate the feasibility of using kinematic GPS together with the existing systems within INDOT for aerial photography acquisition. These techniques are being used by other researchers around the world, and by a few production operations. The factor which made the feasibility questionable in the case of INDOT was their use of a very old RC8 aerial mapping camera. The results of the research indicate that it is in fact feasible to use kinematic GPS within the INDOT photo acquisition and mapping operations. There are several important implementation issues which should be addressed however.

(1) Dual frequency GPS receivers (two minimum) would have to be acquired by INDOT. We used borrowed equipment for our research tests. The receiver for the airplane would not have to be permanently resident there, i.e. it could be removed as needed for other fieldwork operations. Three receivers would be desirable rather than two to provide redundancy. We still have offers from our vendor for loaner equipment for testing purposes. But a production capability would require purchase by INDOT.

(2) The Wild RC8 aerial camera needs to be either retrofitted with an enhanced shutter event signal or replaced with a newer model camera (they are very expensive). We could work with INDOT to make this retrofit, or there are commercial service companies that would do it for a "few thousand" dollars. This is a vital step to bring the accuracy values of the ground points into the decimeter range from the current one meter range.

(3) The much reduced requirement for ground surveys would nevertheless have to be done in a defined coordinate system, such as Indiana State Plane NAD83, rather than local, project-specific systems. This is necessary to be compatible with the geocentric (earth-centered) coordinates produced by GPS.

(4) INDOT stereo mapping personnel would have to become familiar with independent model aerial triangulation. This process involves a measuring task on the stereoplotter prior to the mapping, in which all control points and pass points are marked and measured. This data is then merged with the GPS exposure station data in a simultaneous independent model adjustment. A point marker would have to be acquired. These are available from conventional photogrammetric equipment vendors.

(5) Finally there would have to be some training and collaboration between the authors of this report and the INDOT personnel involved. There would also have to be a close coordination of activities between the photo acquisition operation and the mapping and data reduction operation.

(6) The existing camera should be sent routinely to the US Geological Survey for periodic (approximately 3 year) calibration. This is the usual practice both in private industry and in all other government mapping operations. The self-calibration that we performed for this research was adequate, but it should be supplemented by regularly scheduled calibrations from qualified laboratory.

The driving force behind the implementation of this system must be that other organizations, including several state highway departments are successfully utilizing similar techniques and systems to achieve dramatic productivity gains in their mapping operations. A side benefit of having a survey grade GPS receiver in the aircraft is that it opens the possibility to implement an integrated "flight management system" in which the following of precise flight lines could be automated in areas with sparse landmarks, and further the camera could be driven by the positioning system, rather than the other way around, for such things as quad-centered photography without undue navigation and synchronization burdens on the pilot and photographer.

In summary, there are numerous reasons why this effort should be brought into the production operations within INDOT. The personnel involved in this research would be pleased to collaborate on such an implementation.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Aerotriangulation

The primary purpose of aerial photogrammetry is the compilation of topographic maps. To accomplish this purpose, a stereoscopic model must be created in order to measure the elevation features that are found on a topographic map, such as contours, spot heights, etc. The stereo model requires features identifiable in the stereo pair with known positions on the ground, or control points. A field survey crew establishes these ground points usually at high labor costs and in many cases inadequately. An analytical method used to reduce the need for many costly control points or to solve the problem of inadequate control points is aerotriangulation (Moffitt & Mikhail, 1973). From the ninth chapter in the book, *Manual of Photogrammetry*, (ASPRS, 1980), the following excerpt gives a good introduction to aerotriangulation.

> "Phototriangulation is defined by the Nomenclature Committee of the American Society of Photogrammetry as 'The process for the extension of horizontal and/or vertical control whereby the measurements of angles and/or distances on overlapping photographs are related into a spatial solution using the perspective principles of the photographs. Generally, this process involves using aerial photographs and is *aerotriangulation* or *aerial triangulation*.'"

The image coordinates of the ground points are measured on the photographs. In addition to the ground points, pass points and tie points are measured on the photographs. These points have no known ground coordinates, or object coordinates, however they are chosen so that, in the case of pass points, the same point can be measured on three consecutive photographs, or in the case of tie points, the same points cm be measured on adjacent strips. The pass points and tie points connect the photographs together. Each image point represents a ray of light that "originated" from a point on an object, went through the lens of the camera and ended on the photograph. The set of rays form a bundle of rays that are geometrically defined by the image coordinates of the points together with the camera focal length. The adjustment of all of the bundles in a block of photographs involves the rotation and translation of each bundle in space into such a position that all rays going through the photographic positions of each ground control point will all intersect within measurement error, at the correct object space position. Furthermore, all rays representing each other point, such as a pass point, must intersect at one point in the object space (ibid., 1973).

The condition of collinearity is normally imposed for each ray in all the bundles involved. This condition assumes that if a ray of light is allowed to pass undisturbed through the atmosphere and lens system to a location on the film, this path will be a straight line (ibid. 1973; Lapine,1990). Unfortunately, a ray of light does not pass undisturbed. The image coordinates are distorted by the systematic effects of film deformation, atmospheric refraction, and lens distortion. A mathematical refinement of the images coordinates corrects these distortions (ibid,1990).

GPS

The NAVSTAR Global Positioning System, developed by the U. S. Military since 1973, comprises a constellation of 25 satellites placed in orbits of about 20,200 km above the earth's surface. This constellation provides at least four satellites that are simultaneously visible above the horizon anywhere on the earth, 24 hours a day. Each satellite continuously broadcasts navigation data. There are two carrier waves that are modulated by PRN (pseudorandom noise) sequences in two different codes. The code signals provide a crude method for determining positions in real time by measuring pseudoranges among four satellites and a receiver. Since only the satellite broadcasts the signals, the receiver clock and the satellite clocks are not synchronized. This lack of synchronization between clocks is treated as an unknown in addition to the three coordinates, thus requiring a minimum of four satellites, and thus yielding the term for the distances, pseudoranges. In addition to the broadcast code, phase measurements of the carrier waves provide greater accuracy. Orbit errors, clock errors, atmospheric errors are some of the many errors for which a solution must be found. Post processing techniques using simultaneous observations of several receivers have been developed to solve for these errors.

Static positioning GPS solves for the positions and vectors between stationary GPS receivers. To use GPS while one of the receivers is moving is called kinematic GPS, To solve for the errors in this scenario in a timely manner, one receiver is moving while another receiver is stationary over a known ground point (Seeber, 1993). In addition, phase data should be collected at two different frequencies.

GPS and Photogrammetry

GPS positioning impacts photogrammetry in three main areas. First GPS may be used to control the navigation of the survey flight, thus guiding a pilot along a planned flight path. When the camera is located at the closest distance to a pre-planned position, a computer can trigger an exposure. Secondly, GPS provides high precision camera positions at the time of exposure that are used in aerotriangulation, These camera positions may be used as known control points thus greatly reducing, or in theory eliminating, the need for ground control (Ackermann, 1992; Ackermann & Schade, 1993). Thirdly, GPS may be used indirectly to benefit photogrammetry by greatly reducing the costs of determining the (reduced) network of ground control points. For this project, GPS provided the camera positions at time of exposure. It should be noted that the positions of the ground control used in this project were determined through the use of GPS surveys (Crowl & Merchant, 1995).

There are some known problems and errors which have to be solved when using kinematic GPS positioning of a camera in an airplane. The GPS receiver records ephemeris data emitted by the satellites to determine the position of the receiver. However, what is required is the position of the camera. The GPS antenna offset, or eccentricity vector, has to be measured while the airplane is on the ground. GPS phase measurements have the problem that the initial number of integer cycles, which compose the range to the satellite, is unknown. Known as initial phase

ambiguity, this number can be determined by making stationary recordings before take-off and determining the baseline from the stationary receiver on a reference point to the airplane receiver. The time needed for solving the initial phase ambiguity is about 10 minutes. GPS signal disruption, called loss of lock, spoils the ambiguity solution. Loss of lock may occur when the airplane wings interrupt signals over many seconds. On-the-fly ambiguity resolution techniques avoid the "loss of lock" condition between the moment the airplane leaves a point of known location at the airport and the time the airplane reaches the project area. However, the results of the photogrammetric solution are usually wanted with respect to a local coordinate system. This situation poses a transformation problem which has to be mathematically solved (Ackermann, 1992; Ackermann & Schade, 1993).

INDOT Project

In July 1993, Purdue entered into a research project with the Indiana Department of Transportation (INDOT) to investigate the use of GPS for their photogrammetric projects. This research addressed system operation, calibration, and coordinate accuracy questions. This project implemented and tested GPS equipment and procedures using the INDOT Cessna 206 airplane, and a Wild RC-8 aerial camera, and borrowed (from the manufacturer) GPS receiving equipment.

To refine the image coordinates, the calibration parameters of the camera is required. The U. S. Geological Survey routinely calibrates cameras used in photogrammetry. The camera calibration report contains the calibrated focal length, the radial distortion, the coordinates of the principal points, and the fiducial coordinates. However, the camera used by INDOT had no such calibration report, The project therefore had to determine the camera calibration for the camera installed in the INDOT airplane (Bethel, Johnson & van Gelder, 1993).

CHAPTER 2
LITERATURE REVIEW

Theoretical Results

Using simulated data, Ackermann investigated the theoretical accuracy of GPS blocks for various cases (Ackermann, 1992a). In each case, the block adjustment program was tested as to whether it should have as options (1) no drift correction parameters, (2) one set of linear drift correction parameters for the whole block, and (3) several independent sets of drift correction parameters for each strip in a block. The drift correction parameters model the systematic GPS drift error that has shown up on all experimental tests using kinematic GPS positioning. These efforts seem to be linear in first approximation, at least for time intervals of up to 15 min. (Friess and Heuchel, 1992). In addition to the three options stated above, three scenarios were given as to the placement of ground control. In one scenario, (a), full ground control is placed in the corners of the block. This situation is sufficient to provide a datum transformation from the WGS 84 coordinate system to a local horizontal and vertical system, such as state plane coordinates. In another scenario, (b), full ground control is situated in the corners of the block and vertical control is situated at the beginning and end of each strip. This situation tested what effect drift errors have on the strips. Another scenario, (c), tested the same effect using two cross strips across the ends of the block with vertical control situated at the beginning and end of these cross strips.

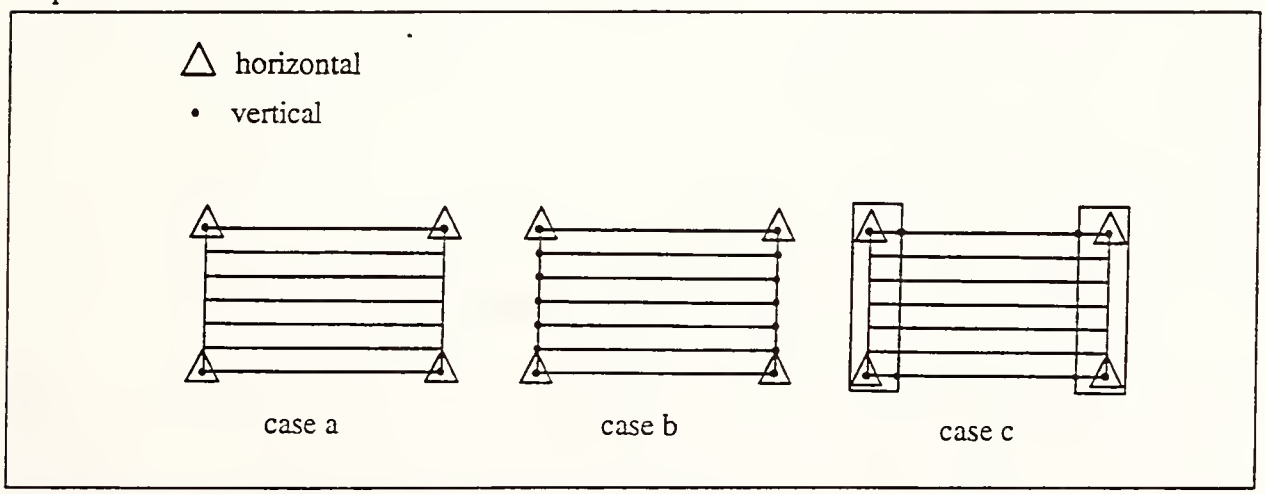


Figure 2.1 Control Scenarios for GPS Block Adjustment

Ackermann first investigated whether block size affects the accuracy. Using control points in each corner of block, a scale of 1:30000, and a $\sigma = 10\ \mu m$ for the image coordinate accuracy, and a $\sigma = 30\ cm$ for control point accuracy and GPS accuracy, three examples were described. These examples showed that the accuracy of GPS blocks is little dependent on block size.

Next, an investigation was conducted to determine whether different drift parameterizations affected the overall accuracy of the block. The results of the investigation are described using

several configurations described in Table 2.1. The a,b, and c refer to the control scenarios in Figure 2.1. Each example used the same block size and photo scale with the GPS control accuracy, $\sigma = 30\,cm$. It is assumed that the uncertainty of the control points and the uncertainty of the GPS determined camera stations are less than the image point uncertainty, scaled to the object space, i.e.

$$s\,\sigma_{im} = \overline{\sigma_{im}} \geq \sigma_{CP}$$
$$s\,\sigma_{im} = \overline{\sigma_{im}} \geq \sigma_{GPS}$$

where s is the reciprocal of the photo scale. In the Table 2.1, the tabulated values are multipliers of $\overline{\sigma_{im}}$, the product representing the expected object space errors in planimetry and height for each configuration. The block size was 6 x 21 photographs.

Configuration	Plan	Height
Case a, no drift parameters	1.0	1.6
Case b, no drift parameters	1.0	1.6
Case a, drift parameters per block	1.7	2.3
Case b, drift parameters per block	1.7	1.7
Case b, drift parameters per strip	2.1	2.3
Case c, drift parameters per strip	1.5	2.0

Table 2.1 Accuracy of Adjusted GPS Bundle Blocks

A last investigation determined how different GPS control accuracies affected the overall accuracy of the block.

In conclusion, Ackermann stated that the theoretical accuracy studies lead to the conclusion that GPS aerial triangulation has highly favorable accuracy features and that the practical application is of greatest operational and economic interest. The use of GPS supported aerial triangulation is highly recommended.

Block Adjustment Model

The solution to the aerotriangulation relies heavily on an iterative method using a computer program called a block adjustment. An important prerequisite for a computer solution is a good mathematical model. The mathematical model must incorporate the eccentricity between the antenna phase center and camera projection center, the GPS drift parameters, and datum

transformation (Ackermann, 1992b; Friess and Heuchel, 1992; Colomina, 1993; Ackermann and Schade, 1993). The eccentricity vector is transformed with rotations ω, ϕ, and κ. The GPS drift parameters have both constant terms and linear terms with respect to time. The datum transformation is accomplished by a linearized seven-parameter transformation. A separate formulation and solution of the datum transfer parameters can be omitted, however, if the approach with linear drift parameters per strip is used, since the corrections automatically include the datum transformation (Ackermann and Schade, 1993). Thus the GPS antenna coordinates are introduced into the combined block adjustment as additional observations for each camera position via the following equations.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{AN} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{PC} + R(\omega,\phi,\kappa) \begin{bmatrix} dX \\ dY \\ dZ \end{bmatrix}_{EC} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} dt$$

Where AN refers to the position of the GPS antenna, PC refers to the projection center (entrance node) of the camera, R is the rotation associated with the exposure, EC refers to the eccentricity vector, and a and b are the drift parameters, both constant and linear with time.

A consideration of the block adjustment is the coordinate system of each of the observables. The image coordinates are measured in a three dimensional Cartesian system (x, y and z). The x and y-axes of the photograph are defined by the fiducial marks, with the principal point taken as the origin. For aerial photographs the x-axis is chosen to be in the direction of the line of flight. The z-axis of the photograph is taken upward to constitute a right-handed coordinate system (Moffitt & Mikhail, 1980). The GPS antenna coordinates are based on an earth fixed Cartesian system (X, Y, and Z) whose origin is the earth's center of mass. The Z-axis is directed toward the conventional terrestrial pole (CTP), the X-axis towards the mean Greenwich meridian, and the Y axis is directed so as to obtain a right-handed system (Seeber, 1993; Torge, 1991). The control points are usually based on a local reference frame, such as state plane coordinate. The state plane coordinate systems were developed by the National Geodetic Survey (NGS) and usually are based on a map projection surface with a two dimensional coordinate system (Eastings, and Northings). The third dimension utilized with the state plane coordinate systems is height, and is based on Mean Sea Level (Moffitt & Mikhail, 1980). The control points may be based on ellipsoidal coordinates (latitude, longitude, and height). Transformations can be made either before the block adjustment or in the adjustment.

Setup Considerations

Before undertaking a GPS project, some practical considerations must be addressed. Many of the considerations are physical, such as placement of the GPS antenna on the airplane; others affect the different measurements needed for the project, such as the eccentricity vector.
The GPS antenna should be mounted on the aircraft in a place where it is free to receive the GPS signals with a minimum of obstruction. If placed directly over the camera on the fuselage, the

phase center of the antenna is located along the optical axis of the camera. This placement minimizes the effect of the crab angle on the eccentricity vector, but may result in multipath (the reception of reflected signals) problems. A placement of the antenna on the fuselage may also produce shadowing of the signal as the airplane makes turns resulting in loss of lock. Placing the antenna on the vertical stabilizer reduces the possibility of multipath or shadowing, but complicates the measurement of the eccentricity vector (Merchant, 1993; Curry & Schuckman, 1993). The eccentricity, the offset vector between the antenna phase center and the projection center of the camera, is measured using close range survey techniques. Observations to the fiducial marks and the antenna are made using theodolites, distance measurements, and leveling. The distance from the focal plane to the entrance nodal point in the lens is also determined. While the measurements are being made, the camera should be placed in the zero kappa position, Later during the flight, the attitude angles, especially the crab angle, should be recorded (Jacobsen, 1993; Curry & Schuckman, 1993; Schuckman; Curry, & Salsig, 1992; Merchant, 1993).

The connection between the camera and the receiver records the event of the lens opening in the receiver. Modern aerial cameras have an output to the receiver to record the center time of exposure. Older cameras can use a photo diode in the image plane. The recorded time of the diode has to be calibrated relative to the instant of exposure as a function of the exposure time (Curry & Schuckmann, 1993; Jacobsen, 1993). The GPS receiver records data at a fixed interval. The time of the camera exposure is normally linearly interpolated between these intervals. Kalman filters and Lagrange interpolation have also been used to interpolate the time of camera exposure. The desired recording rate for the GPS receiver should be no less than 60 per minute, since this represents approximately 60 m at a ground speed of 200 km/h (125 m/h.) A faster recording rate may fill up memory too quickly (Ackermann 1991a; Ackermann 1993; Jacobsen, 1993). Another possibility would be to delay the camera event to nearly coincide with the receiver. Some recent cameras have the ability to cause an exposure within 50 milliseconds from the time of request (Merchant, 1993). The actual mission should be planned in accordance to how the satellite constellation appears at a particular date and time and location. This may be accomplished using the planning software supplied by the receiver manufacturer. The Positional Dilution of Precision (PDOP) is determined by satellite geometry. The PDOP can be interpreted as the reciprocal of the volume of a tetrahedron that is formed from the satellite and user positions. The best geometric situation exists when the volume is maximized, and hence when PDOP is minimized (Seeber, 1993; Curry & Schuckman, 1993).

The placement of the base receiver should be located where multipath and obstructions are minimized. The location should be over a known control point within 500 km from the mission area (Curry & Schuckman, 1993; Ackermann, 1992a; Ackermann & Schade, 1993). To solve the initial ambiguity, both receivers make stationary recordings before take-off and determine the baseline from the stationary receiver to the airplane receiver. The procedure used to take about an hour, but with the fast ambiguity solutions now available, 5 to 10 minutes is adequate (Curry & Schuckman, 1993).

Practical Results

A summary of 23 GPS block projects shows how well empirical results compare with theoretical results. These GPS block projects were carried out only by the Inpho company. All adjustments were successful, although the data were obtained from different companies and refer to different GPS receivers. The projects came from various countries and cover a wide range of photo scale (from 1:6100 to 1:50000). The block sizes varied between 12 photos and 1633 photos per block or strip. A variable number of cross strips was used, depending on shape and geometry of a block. Although there were individual deviations, the average r.m.s. accuracy results, as derived from 9 blocks with check points, were $2.5\sigma_o s$ for x, y, and $2.2\sigma_o s$ for z as compared to the general theoretical expectation of $2.5\sigma_o s$ for x, y, and $2.0\sigma_o s$ for z (Ackermann & Schade, 1993; Ackermann, 1994).

Lapine conducted a project at the Transportation Research Center near Marysville, Ohio in 1988 (Lapine, 1992). The photography was flown at a height of 6000 feet resulting in a photo scale of 1:12,000. Two different flight missions were flown using two different RC-10 cameras resulting in three strips of 5 photos each. One camera was equipped with a reseau while the other camera was equipped with eight fiducial marks. The GPS data was collected using three Trimble 4000 SX, 5 channel receivers. These receivers are of the single L1 frequency phase comparison type. Both spatial and time offsets were carefully measured, and an aerial calibration of the camera was performed. The known ground control positions were compared against the computed ground control positions using GPS. The results are tabulated for the aerial calibrated camera and the conventional camera calibration for both the reseau equipped camera and the fiducial camera. The author attributed the poorer results of the fiducial equipped camera to the effect of non-linear film deformation which cannot be adequately modeled by the eight fiducial marks.

	Aerial Calibration			Conventional Calibration		
	X	Y	Z	X	Y	Z
Mean	-0.001	0.005	-0.065	0.081	0.090	0.704
σ	0.033	0.028	0.085	0.371	0.429	0.180
RMS	0.032	0.028	0.106			

Table 2.2 Reseau Equipped Camera

	Aerial Calibration			Conventional Calibration		
	X	Y	Z	X	Y	Z
Mean	0.047	0.007	-0.081	0.101	0.274	2.427
σ	0.059	0.064	0.131	0.508	0.602	0.265

RMS	0.075	0.063	0.151			

Table 2.3 Fiducial Equipped Camera

Friess and Heuchel describe the results of a project flown over Glandorf in Germany (Friess and Heuchel, 1992). The block covers 5 strips in the N-S direction with 14 photographs each and 7 cross strips in the E-W direction with 10 photographs each. The photo scale is 1:8000, with forward overlap of 60% and side overlap of 20%. The aerial camera used in the project was a Zeiss RMK TOP, which is able to generate a pulse at the moments of exposure. The data was collected at a rate of 0.5 sec with two Ashtech L-XII receivers. The test area contained 15 full control points and 20 vertical control points.

The coordinates of the projection centers were determined independent of the GPS positions by a bundle adjustment. The spatial distance between the GPS antenna center and the camera projection center was calculated. The distances between the two centers is not affected by the variations of aircraft attitude. The variation of this distance within the individual photo strips was taken as criteria for the assessment of the accuracy of the GPS aircraft antenna positions. The RMS values of the difference between the distances varied between 8.6 cm. and 37 cm. Correction for linear drifts per strip were applied and the corresponding RMS values of the difference between the distances varied between 4.1 cm and 8.4 cm.

The authors also described an analysis of two different block configurations. The first configuration, 5 + 2, consisted of the five strips in the N-S direction and two cross strips along the ends. The second configuration, 7 + 2, consisted of the seven strips in the E-W direction and two cross strips along the ends. For both GPS blocks, only 4 control points situated in the corners were applied. The RMS values of the differences of the adjusted coordinates and the given coordinates of the check points in the adjustment were computed as the empirical accuracy. The theoretical accuracy is derived from the inversion of the normal equation system. The results, in centimeters, are tabulated below in Table 2.4. The authors attribute the poor quality of the photographs due to unfavorable weather conditions for strips 7 and 11 for the slightly worse accuracy of the 7 + 2 block.

Block	Empirical Accuracy		Theoretical Accuracy	
	RMS v (xy)	RMS v (z)	σ (xy)	σ (z)
5 + 2	6.3	8.5	4.4	9.7
7 + 2	7.8	10.8	4.3	9.2

Table 2.4 Accuracy of GPS-supported Block Adjustment

Gruen, Cocard and Kahle describe the results of a project flown over Uster near Zurich, Switzerland (Gruen, Cocard & Kahle, 1993). Eight strips with a forward overlap of 80% (with only 60% used for processing) and side overlap of 60% were photographed using a Wild RC 20

aerial camera. The flight was flown at a height of about 1500 m above the ground, resulting in an image scale of 1: 10,000. Three Trimble SST GPS receivers were used in the project, one positioned at a reference site near the airport, the other two were mounted on the airplane, The reference receiver and one of the receivers on the plane were dual frequency receivers and collected data at a rate of 1.0 per second. The other receiver was a single frequency receiver and collected data at a rate of 2.0 per second. The block contained 94 control points. The RMS values of the differences between the GPS computed camera projection centers and the bundle adjustment computed camera projection centers gave $\mu_x = 27.2$ cm, $\mu_y = 74.6$ cm, and $\mu_z = 31.5$ cm. When shift parameters for the entire block were included the RMS values of the differences became $\mu_x = 12.7$ cm, $\mu_y = 17.4$ cm, and $\mu_z = 7.5$ cm. The authors report better values if the shift parameters are computed for each strip, but did not show the results. Another result of the report compared various scenarios using different numbers of ground control with the GPS control. The scenarios include using all the control available for the block, dense control, four full control points in the block corners with a vertical control point in the center, and finally four full control points in the block comers. An interesting result of the report shows that the accuracy values using four control points in the block comers with GPS ($\mu_{xy} = 5.2\,\mu m$, $\mu_z = 8.7\,\mu m$) are about the same as the accuracy values using dense control without GPS ($\mu_{xy} = 5.0\,\mu m$, $\mu_z = 8.2\,\mu m$).

Jacobsen describes the results of three GPS projects flown over Germany (Jacobsen, 1992; Jacobsen, 1993). the results of only one of the projects are included here, since for this project, the author takes a different approach with the data. This project was flown over Blumenthal in August of 1988 in 5 strips resulting in 69 photos. The endlap was 80% and the sidelap was 60%. The photo scale was 1:6300. The project used Trimble TI 4100 receivers at a time interval of 2.2 seconds. The block contained 69 control points. A bundle adjustment was computed without the GPS-positions as additional observations. The resulting RMS differences in the projection centers determined by GPS (after eliminating systematic errors) are RMS-X = 18 cm, RMS-Y = 19 cm, and RMS-Z = 10 cm. A comparison was made between how overlap changes the accuracy (p = endlap, q = sidelap). The following tabulations give the results using four control points and one control point (in centimeters).

	No GPS			With GPS		
	Sx	Sy	Sz	Sx	Sy	Sz
p=80%, q=60%	12.0	6.1	17.5	6.2	6.2	12.8
p=60%, q=60%	12.2	6.8	22.9	10.5	4.5	9.4

p=60%, q=20%	26.5	10.1	36.0	15.2	8.2	12.5

Table 2.5 Results using Four Control Points

	Sx	Sy	Sz
p=80%, q=60%	8.5	14.1	24.0
p=60%, q=60%	19.0	22.8	27.0
p=60%, q=20%	22.7	27.1	29.6

Table 2.6 Results for One Control Point with GPS

Schuckman, Curry, Zhao, and Salsig describe the results of a project flown near Yosemite National Park in California (Schuckman, Curry, Zhao, & Salsig; 1992). The elevation of the project area ranged between 2100 feet to 8800 feet above sea level. All flight lines were oriented in the north south direction. The project produced four flight lines of five exposures each with a 60% endlap and a 40% sidelap. The photo scale was specified at 1:40,000. A Trimble 4000 SST GPS receiver was connected to the Zeiss RMK TOP camera during the flight. The camera sent a signal to the receiver at the midpoint of each exposure, The data collected were dual frequency carrier phase observations at a 2.0 second rate. The carrier phase observations were processed as well as the pseudorange observations. Four airborne-controlled bundle adjustments were computed, two with kinematic camera stations and two with pseudorange camera stations. The coordinates of 79 triangulated pass points were compared to the convention ground controlled adjustment. The tabulation of the results (in meters) follows for the kinematic camera stations and the pseudorange camera stations.

	No Ground Control Points			Four Ground Control Points		
	dX	dY	dZ	dX	dY	dZ
mean	-0.4	-0.4	-0.8	-0.1	-0.2	-0.3
RMS	2.8	2.8	2.5	2.1	2.2	2.2

Table 2.7 Kinematic Camera Stations

	No Ground Control Points			Four Ground Control Points		
	dX	dY	dZ	dX	dY	dZ
mean	-2.5	-0.1	-0.4	-0.9	0.1	0.2
RMS	2.4	2.4	2.4	0.5	0.6	2.1

Table 2.8 Pseudo-Range Camera Stations

CHAPTER 3
PROPOSED METHODOLOGY

The next four chapters are entitled System Design and Configuration, Data Acquisition, Data Analysis, and Conclusions and Recommendations. The following paragraphs give a quick overview of the sections in these chapters.

In System Design and Configuration, three operations are described: the design and testing of the interface between the camera and the receiver, the mission planning, and the design and configuration of the equipment to measure the shutter delay. In Data Acquisition, four operations are described: a summary of the activities on the day of the GPS-photogrammetry field test, the measurements to determine the eccentricity vector between the antenna phase center and the camera entrance node, the measurement of the image points and some refinement to determine the fiducial coordinates, and the measurement of the shutter delay.

In Data Analysis, six operations are described: the GPS post-processing and results, the determination of the eccentricity vector, the determination of the camera calibration, the determination of the shutter delay, the block adjustment without GPS, and the block adjustment with GPS.

Since the strip is overly controlled, a good solution to the camera parameters is possible. This solution is used to compare with the GPS derived camera locations. Various scenarios using little control have been tested by placing the control points in different locations within the strip to compute how the placement of control affects the accuracy. Conclusions and recommendations about the results of the project are made in the final chapter.

CHAPTER 4
SYSTEM DESIGN AND CONFIGURATION

Interface

The GPS receivers are capable of recording an externally triggered event when a TTL pulse is sensed. The intervalometer sends a +28 volt pulse to the shutter to begin the sequence for opening. An interface between the intervalometer and the GPS receiver was built so that when the +28 volt pulse was sent to the shutter, a reed relay was also closed causing a ground to be read by the receiver. The receiver interpreted this as an event to be recorded in memory.

A receiver antenna was installed on top of the fuselage and wing of INDOT's Cessna 206 aircraft. The location of the antenna was chosen to minimize the effect of the attitude of the airplane during flight. To test the interface, a flight was made at Eagle Creek Airport using a dual-frequency Ashtech P-XII GPS receiver in the airplane and two single-frequency receivers on the ground. The photo signals could not be recorded properly during this flight test however, and a second test nine days later proved very successful. When the data was analyzed, a few double events were recorded. It was determined that the second event was an echo of the first event and could be ignored. The reason for the echo was not investigated, however it was thought to be caused by a double bounce on the reed relay (Bethel, Johnson, and van Gelder, 1995).

Mission Planning

The Ohio Department of Transportation maintains calibration and test fields in Madison County Ohio. The calibration range consists of 48 targets spaced along three major east-west roads, Interstate 70, State Route 40, and Old Columbus Road. Targets were also found along Potee Road which runs diagonally from northwest to southeast and crosses the three roads. The calibration site is approximately 1.37 miles east and west and 1.06 miles north and south. The test range is located just east of the calibration range and consists of 104 targets spaced along State Route 40 and Interstate 70 for a total east-west distance of 8.5 miles. Control points were also maintained along three roads crossing between State Route 40 and Interstate 70. They are State Route 56, Gwynne Road, and State Route 36 (Crowl, Merchant, 1995).

The mission was planned to meet two objectives. First, since the camera had no calibration report, the mission would require a method to recover the calibration parameters. The calibration range was used for this purpose. Secondly, the mission would photograph a strip along the entire stretch of the calibration and test ranges. These photos would be used to check the accuracy of the GPS controlled aerotriangulation and to investigate systematic errors.

For the calibration portion of the flight, the mission plan called for six photographs, two vertical photographs and four oblique photographs taken in each of the cardinal directions. The oblique photographs were planned to be taken at a tilt angle of 35 degrees and an altitude of 6500 feet

above sea level (1987 meters). For the strip portion of the flight, the mission was planned for an altitude of 4400 feet above sea level (1341 meters). The average height of the land in the flight area is about 980 feet above sea level (300 meters). The forward overlap for the photographs was set at 60%. It should be noted that in actuality the average flying height above the ground for the calibration portion was 1640 meters (5380 feet). This produced a nominal scale of 1: 10800, and the length of each photograph on the ground was 2480 meters (8136 feet). The oblique photographs were taken at tilt angles of 18, 21, 22 and 25 degrees. The average flying height above the ground for the strip portion was 1056 meters (3464 feet). This produced a scale of 1:6930, and a length of each photograph on the ground was 1594 meters (5230 feet). The 60% forward overlap produced a nominal base distance of 638 meters (2092 feet) on the ground.

Shutter Delay

The camera used by the Indiana Department of Transportation is a Wild RC-8. The rotary shutter requires a +28 volt pulse from the intervalometer. Once this pulse is received, a sequence of events results in the opening of the shutter. Depending on the location of the rotating parts of the shutter, the time from the pulse to the effective shutter opening will vary. Thus the shutter delay should be uniformly distributed. A method was developed and tested to determine this delay.

A simple circuit was built consisting of a resistor and an infrared photo transistor. By choosing the size of the resistor, the circuit could either be light sensitive or have a fast reaction time. The resistor was chosen to produce a fast reaction time, (about 1 μs). When light was present the photo transistor conducted causing ground to be produced at the output. When light no was present the photo transistor lost its ability to conduct and a voltage could be measured across the transistor. The photo transistor was placed for testing and development in the housing of a 35 mm camera and covered up to keep any extraneous light. An oscilloscope was hooked up to the circuit so that any signal could be seen. When the camera lens opened, a signal was seen on the oscilloscope.

A method was required to measure the time it took for the shutter to open once the signal was sent from the intervalometer. A 25 MHz, 8 bit analog to digital converter (ADC) was used. The ADC could digitize an analog signal between -5 volts and +5volts. The sampled signal could either be written directly to the computer's main memory or stored in a 4 KB cache. An acquisition rate could be set by the user if the cache is used; however, if main memory is used, the acquisition rate is set at 25Mz and cannot be changed. The unfamiliarity with the ADC and the desirability of controlling the acquisition rate prompted the use of the cache. The shutter speed for the test was planned at 1/500 of a second, or equivalently, 2 ms. To get a good shape of the shutter opening, a minimum of five samples were required. Thus each sample should represent 0.4 ms. With these requirements, the sampling rate was determined to be 2500 Hertz. It would take 1.6 seconds to fill the cache at this rate.

The next requirement was to keep the ADC from digitizing the input signal until needed. An

external Input-Output connector includes a pin which acts as an Inhibit/Release. While the pin is pulled low, the ADC still runs, but the address generator and sample counter are inhibited and data storage is halted. This allows synchronizing the data acquisition with external events. When +5 volts are applied to the pin, the cache begins to store the input signal. To keep +5 volts to the pin long enough for the entire signal to be digitized, a timing circuit was built. This timing circuit receives the signal from the reed relay and allows a +5 volt signal to be applied to the pin for a period of 2 seconds (Silicon Alley, 1989).

A demonstration Turbo Pascal program supplied by the manufacturers of the ADC was modified so that the acquisition frequency could be changed and the data written to a file. This setup was tested and found to work properly with a 35 mm camera and the photo transistor.

Figure 4.1 shows the diagram of the RC-8 shutter timing circuit.

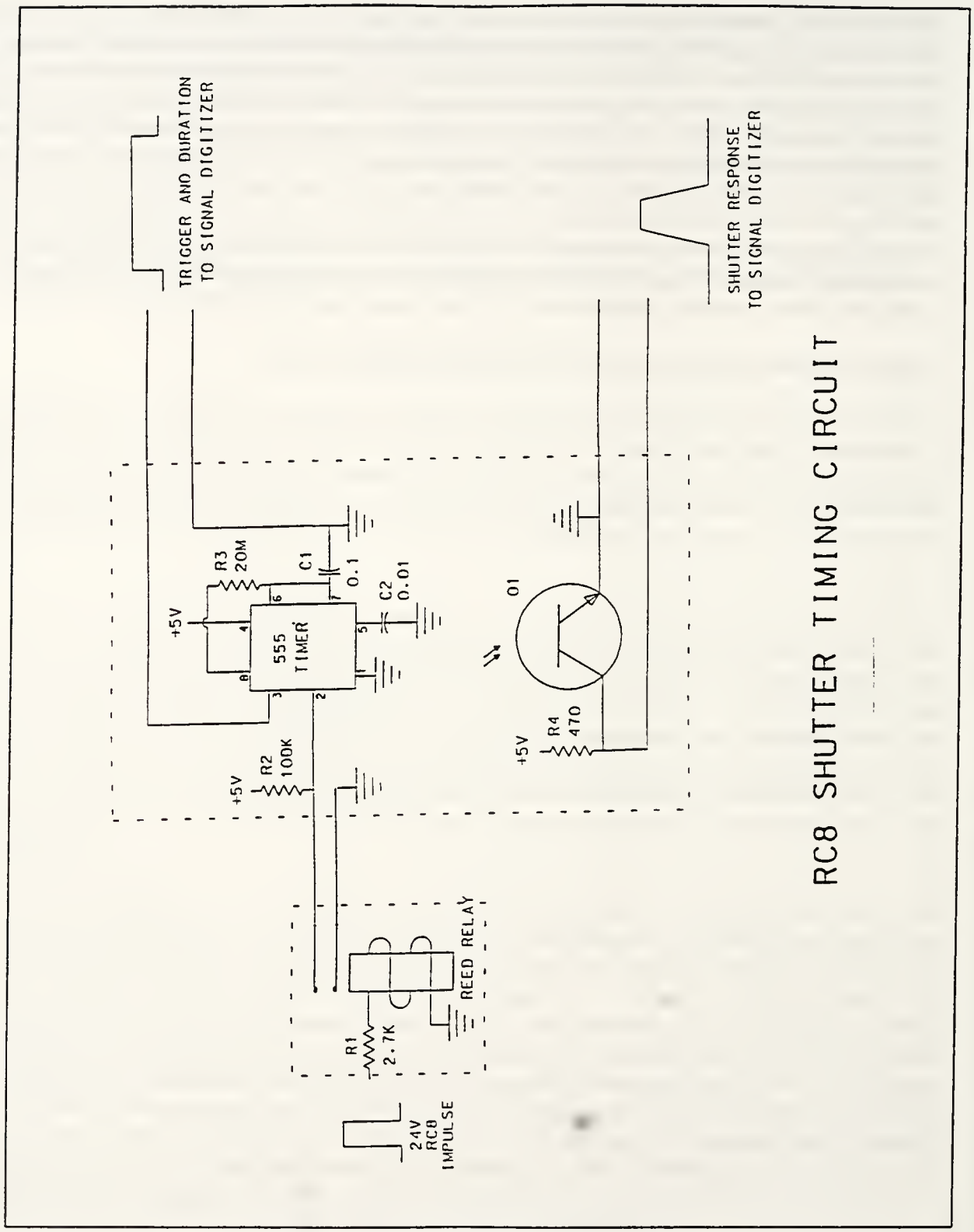


Figure 4.1 RC-8 Shutter Timing Circuit

CHAPTER 5
DATA ACQUISITION

GPS-Photogrammetry Field Test

Professors Johnson, van Gelder, and Bethel, along with approximately seven graduate and undergraduate students met at 6:00 AM on 20 April, 1994 in West Lafayette and drove to Madison County Airport, Ohio, arriving at 9:30 AM. All times are Eastern Standard, even though Ohio was on Eastern Daylight at the time. James Kinder, pilot, and Dave Peipho, photographer, arrived at Madison County Airport at around 10:30 in the INDOT Cessna 206, having flown from Eagle Creek Airport in Indianapolis. The weather was clear.

Three Dimension Ashtech single frequency receivers, an Ashtech M-XII single-frequency receiver, and three Z-XII Ashtech dual frequency receivers were available for the project. An Ashtech Z-XII dual frequency receiver occupied an established ground control station adjacent to (west of) the airplane parking area (designated as MAD1). Another Ashtech Z-XII dual frequency receiver was placed inside the Cessna photo aircraft (designated as TST3). Initial inspection of the GPS constellation configuration showed that there were eight satellites in view, yielding favorable conditions for the field test. Personnel from ODOT, Ohio Department of Transportation, were there at the same time (performing similar operations) and it was determined that the proximity of their Rogue GPS receiver was interfering with reception by the Ashtech units. They moved their equipment so that no further problems were encountered of this kind. This Ashtech equipment was on short-term loan from the Ashtech company, which had agreed to such a research support contribution following Purdue University's purchase of some of their Dimension units, A third Z-XII was shipped on loan, but it was unable to be used because of a malfunction, Of the two units which were working, only one seemed to accept the "photogrammetry" input signal, therefore there was regrettably a lack in redundancy of equipment (and later in data!).

An Ashtech Dimension receiver (designated as POTE) was set up over the ODOT control point on the Potee Road overpass over I-70. Note that this was not a photo targeted point. A second Dimension receiver (designated as OH38) was set up over the ODOT control point on the Ohio State Road 38 overpass over 1-70. A last control point was occupied within 100 meters of MAD1 using the Ashtech M-XII receiver (designated as EIGH). The third Dimension receiver was used as a backup. Communication with the students at Ohio 38 was maintained by radio. Communication with the Potee Road crew was by vehicle shuttle. The plane was manually positioned over the orientation (compass) rose on the tarmac in the parking lot just north of the taxi way. All receivers were turned on at about 11:30, and the plane occupied the orientation rose station for about 15 minutes. Professor Johnson rode in the airplane along with the pilot and the photographer. Takeoff was to the west on runway "26" (i.e. 260 degrees magnetic azimuth) at about 11:45 am.

The plane climbed to about 2600 meters altitude (MSL) and flew the requested pattern over the densely targeted area around Potee Road, Columbus Road, National Road, and I-70. This pattern

consisted of four nominally vertical photographs centered over the "Z" road pattern of Columbus, Potee, and National Roads. Likewise four low obliques were taken from the four cardinal directions of the same area. It was requested to obtain as much tilt as the occupants of the plane could stand, and they managed 20 to 25 degrees for the four obliques. These were needed to provide geometric strength for the later estimation of inner orientation parameters which are not able to be recovered from nominally vertical photographs over flat terrain.

Having completed the "calibration" block, the plane then descended to 1300 m. and flew along the path eastward covering I-70 and National Road, both heavily targeted with control points. The photographer was asked to not move the camera once they came onto the flight line. Unfortunately, the crab angle was not recorded. This proves to be an important parameter in GPS-photogrammetry since it affects the absolute orientation of the eccentricity vector between the antenna and the camera entrance node. Approximately 40 exposures were made in total. Subsequently 8 photographs were used for the self-calibration, and 17 for the strip adjustment tests. The plane landed at about 12:30 and taxied over to the orientation rose. The plane was manually pushed so that it was facing south with the camera viewfinder centered over the center of the orientation rose. It was left in this position for about 15 minutes. This was just a precaution, as recent advances in GPS processing techniques permit the recovery of the integer ambiguities on the fly. Inspection of the receiver on board the plane indicated that the shutter events were apparently being correctly recorded in the receiver.

Finally a dimension receiver was set up over the orientation rose and the station was occupied for about 45 minutes. All the people and equipment were collected and departed the airport for home at about 2:30.

Figures 5.1 - 5.3 illustrate the location of the photographs for the strip along the ODOT calibration and test range. The first figure gives an overview, while the last two figures show greater detail.

Eccentricity Vector

To determine the eccentricity vector between the fiducial center and the antenna phase center of the receiver, two theodolites were placed near the airplane. Azimuth and zenith measurements of sixteen targets were observed from the two theodolites. A meter bar was placed between the door of the airplane and the floor, three targets were observed at 0 cm, 50 cm, and 100 cm marks. Four targets were placed arbitrarily on the chassis of the airplane, three targets were placed arbitrarily on the floor inside the airplane. Since the antenna could not be observed directly from the theodolite, a 15 cm ruler was placed on the receiver with 0 cm above the antenna. Two targets were observed on this scale at the 5 cm and 10 cm marks. The side fiducials were used as the last four targets. No attempt was made to orient or level the airplane.

To determine the distance between the fiducial center and the entrance node of the camera lens, a drawing of the lens system was obtained (Burnside, 1985). The distance between the entrance node and the exit node was determined by scaling the drawing. This method was employed by

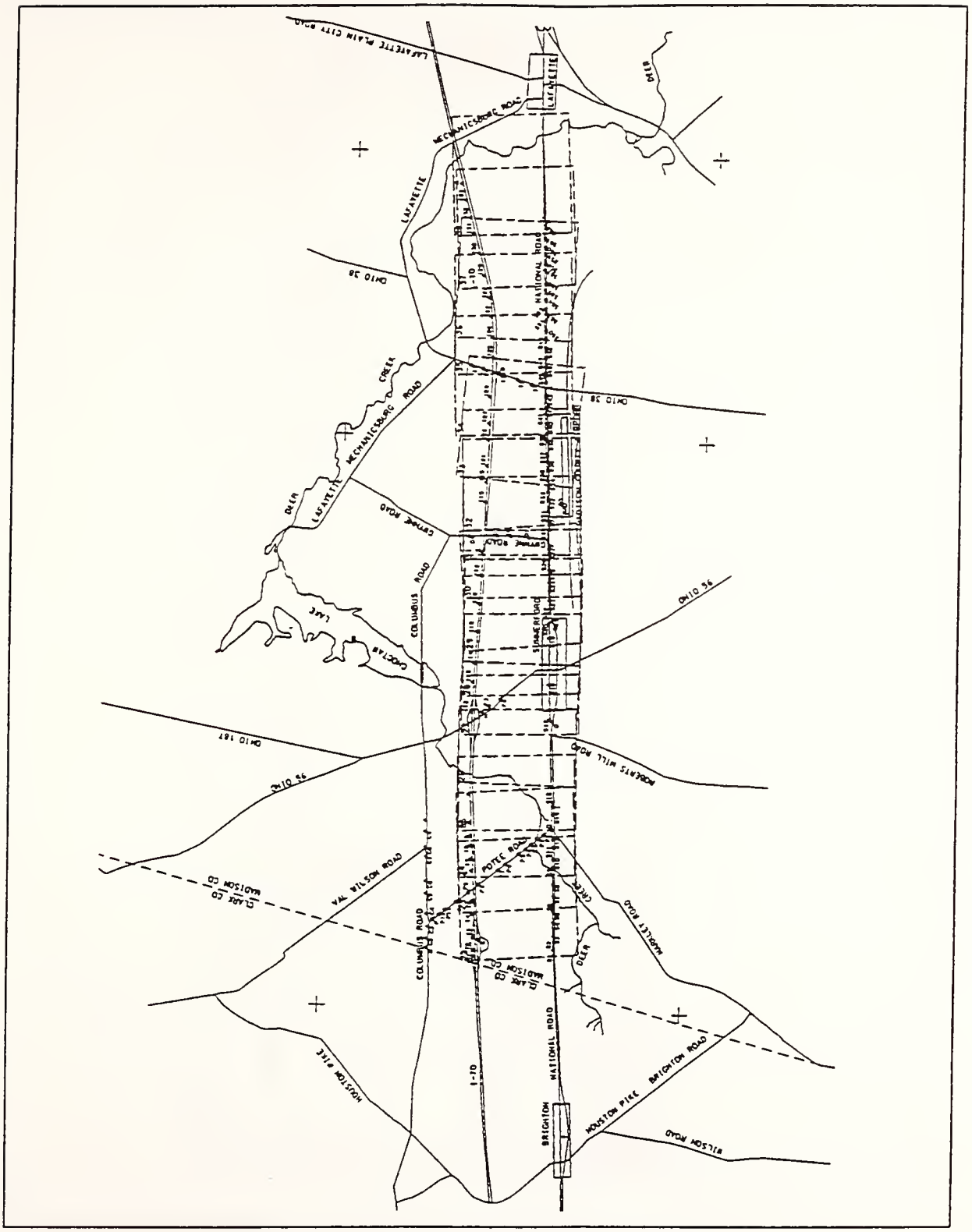


Figure 5.1 ODOT Test and Calibration Range

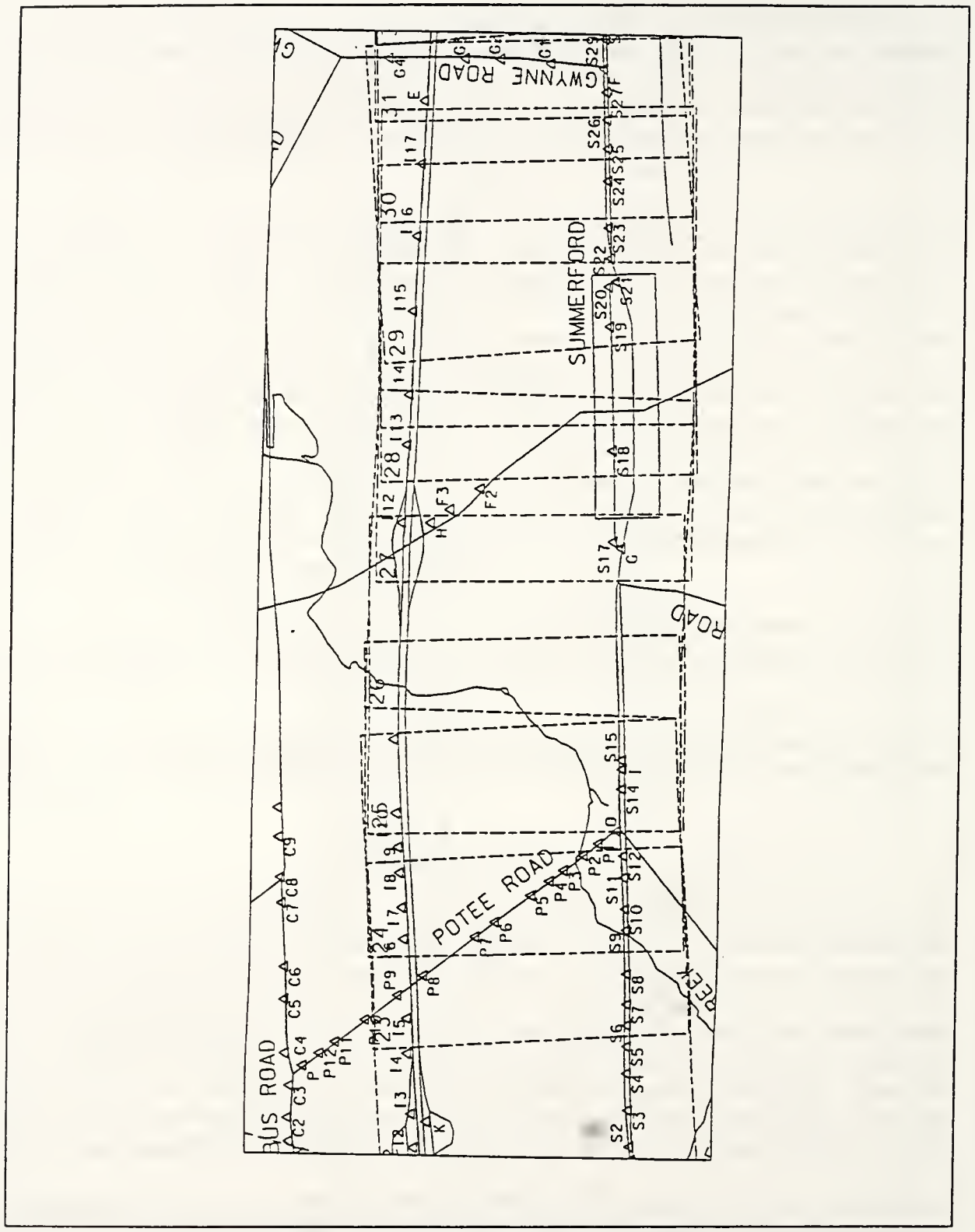


Figure 5.2 ODOT Test Range (west end)

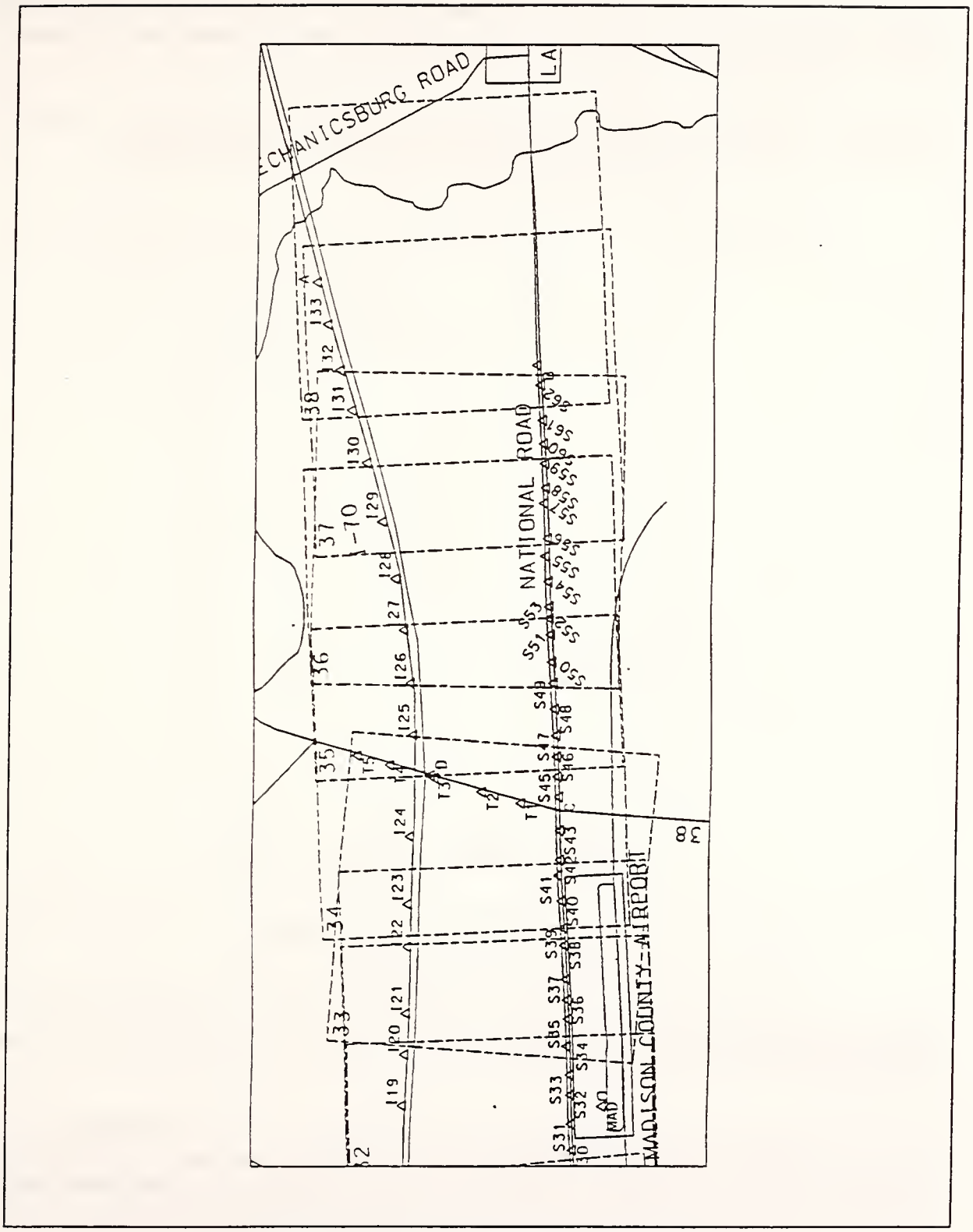


Figure 5.3 ODOT Test Range (east end)

Lapine (1991). This method produced an internodal distance of 0.113 meters. Thus the distance from the focal plane to the exit node is the sum of the internodal distance and the nominal focal length, 0.113 + 0.152 = 0.165 meters.

Figure 5.4 illustrates the location of the camera, the GPS antenna, and the placement of the theodolites. Figure 5.5 shows the internodal distance needed for the calibration.

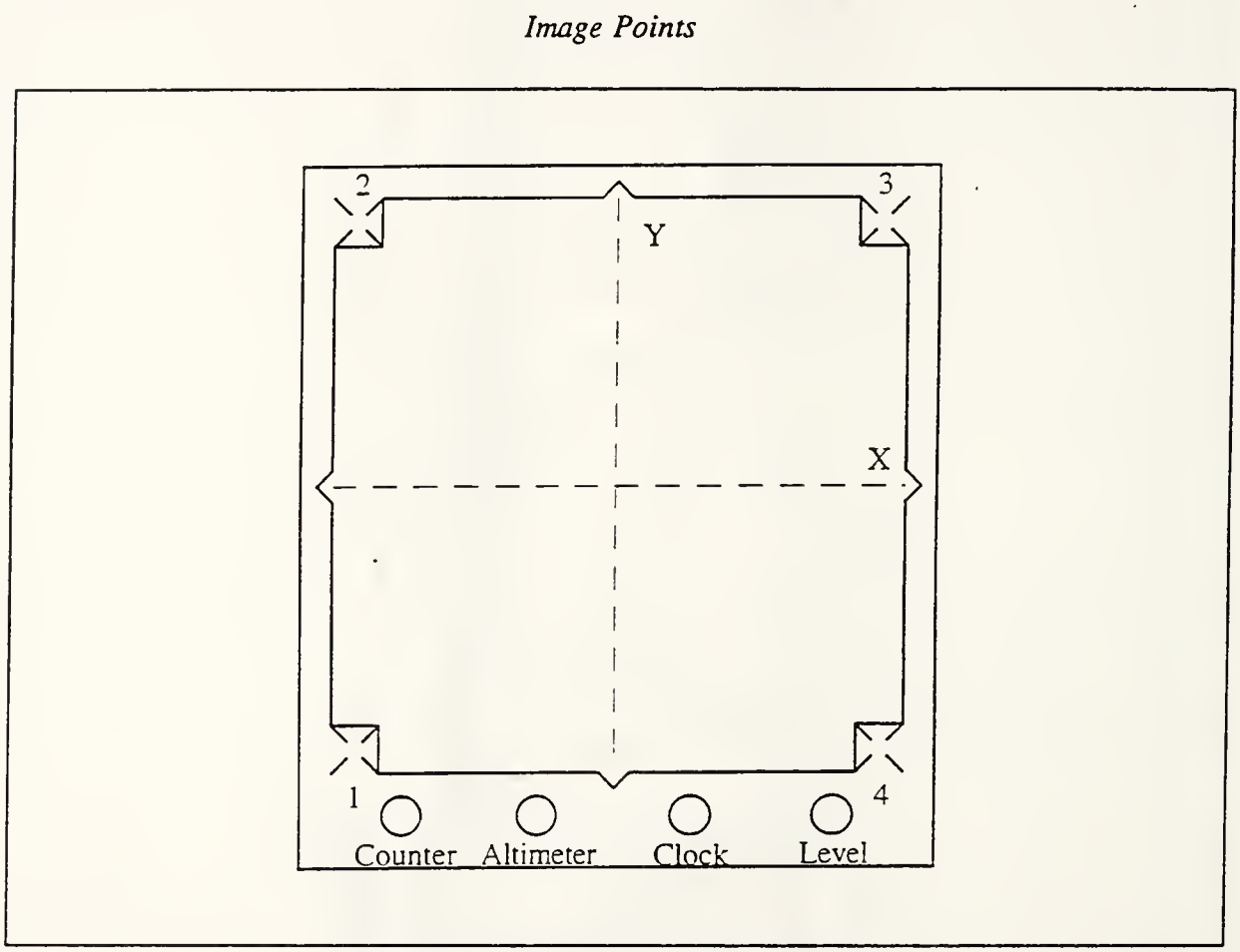


Figure 5.6 Location of Fiducial Marks

Before measuring the images, the fiducial order was determined. With the data strip on the bottom and starting in the lower left corner, the fiducials are numbered clockwise, as illustrated in Figure 5.6.

Due to the shape of the fiducial marks, the fiducial center could not be measured directly. By starting in the lower left corner, the fiducial ticks are numbered clockwise, as illustrated below. The fiducial tick is measured at the inside edge, with the measuring mark just inside, see Figure 5.7. Thus sixteen measurements are required to determine the location of the four fiducials.

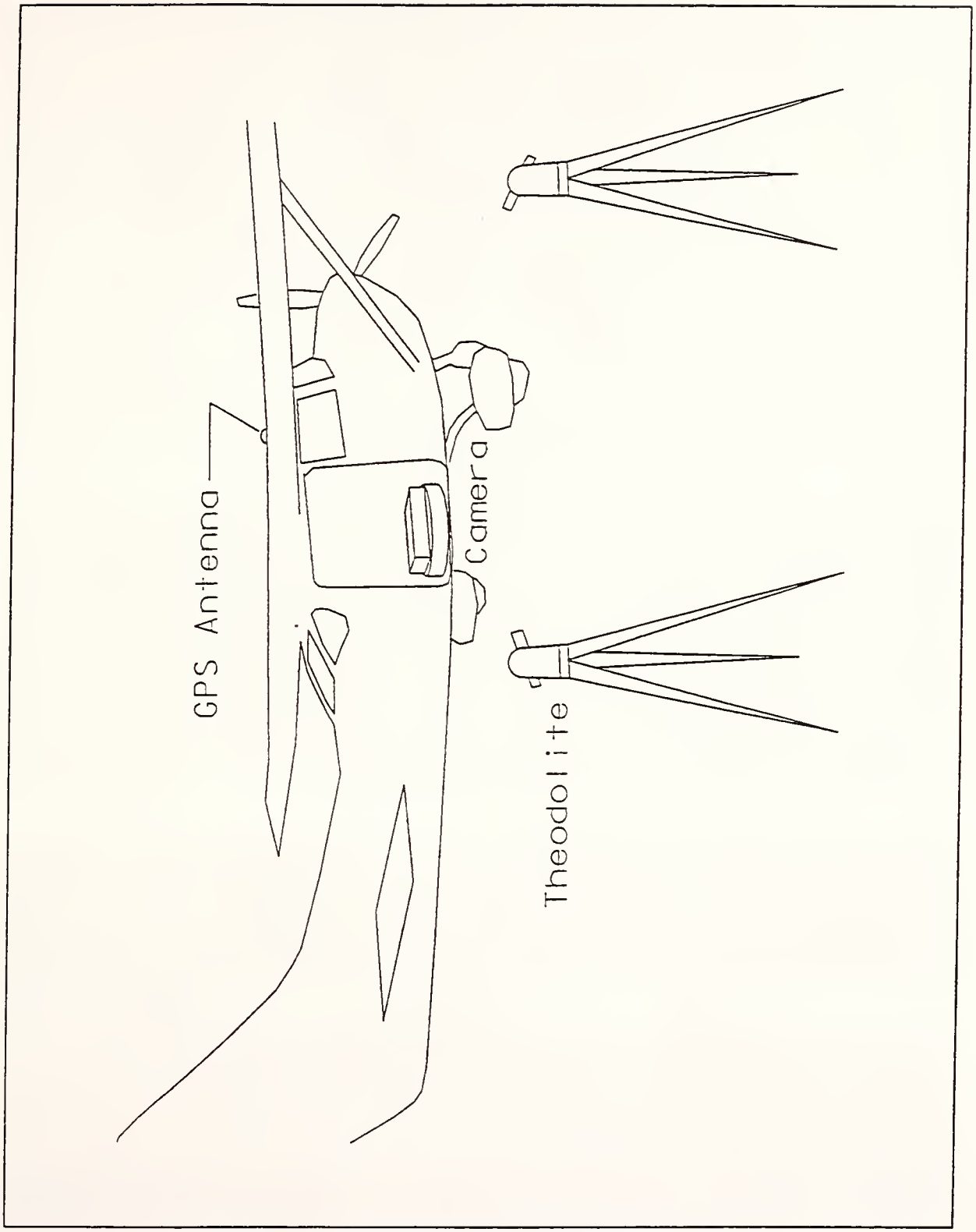


Figure 5.4 Measurement Setup for Eccentricity Vector

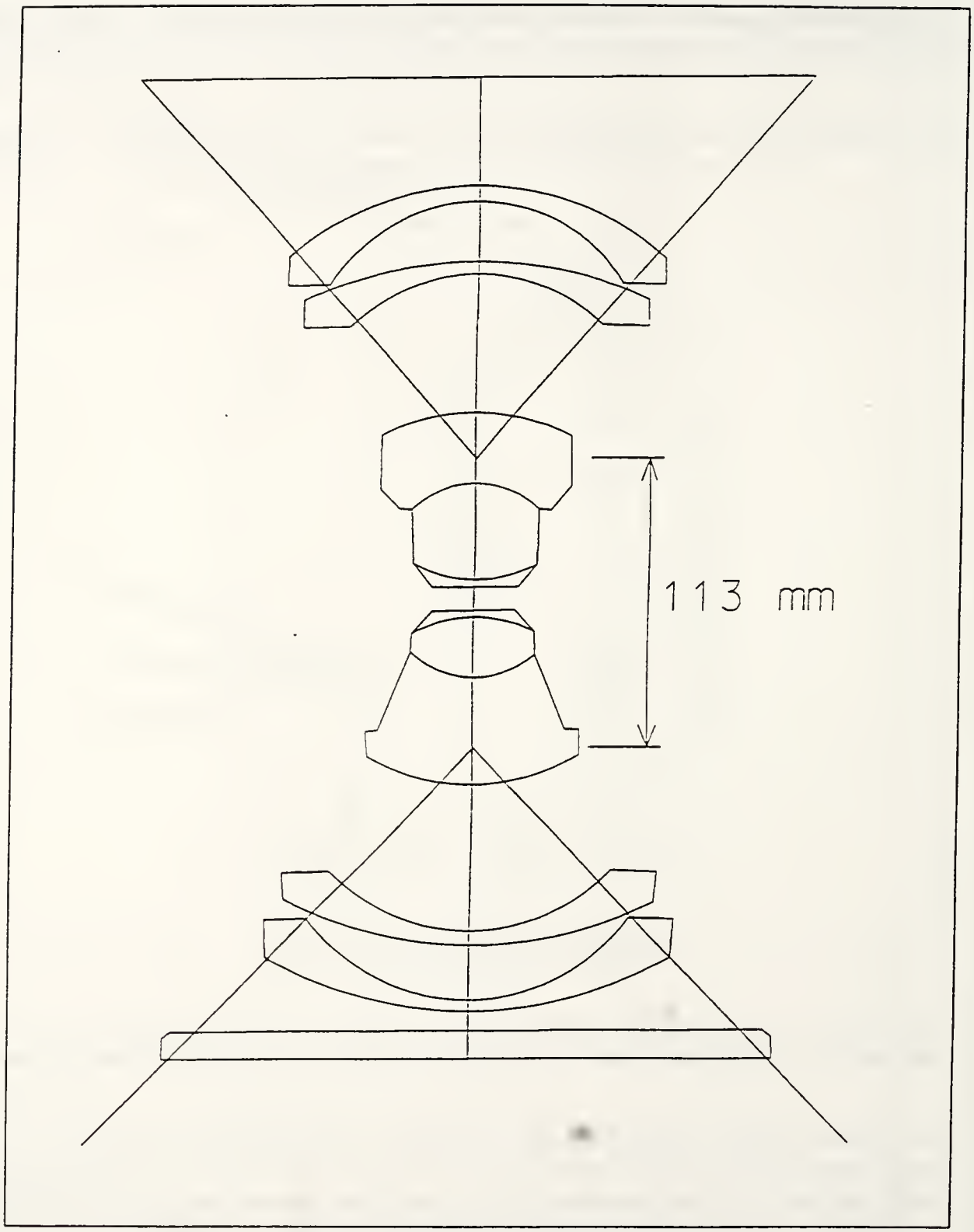


Figure 5.5 Wild RC-8 Aviogon Lens

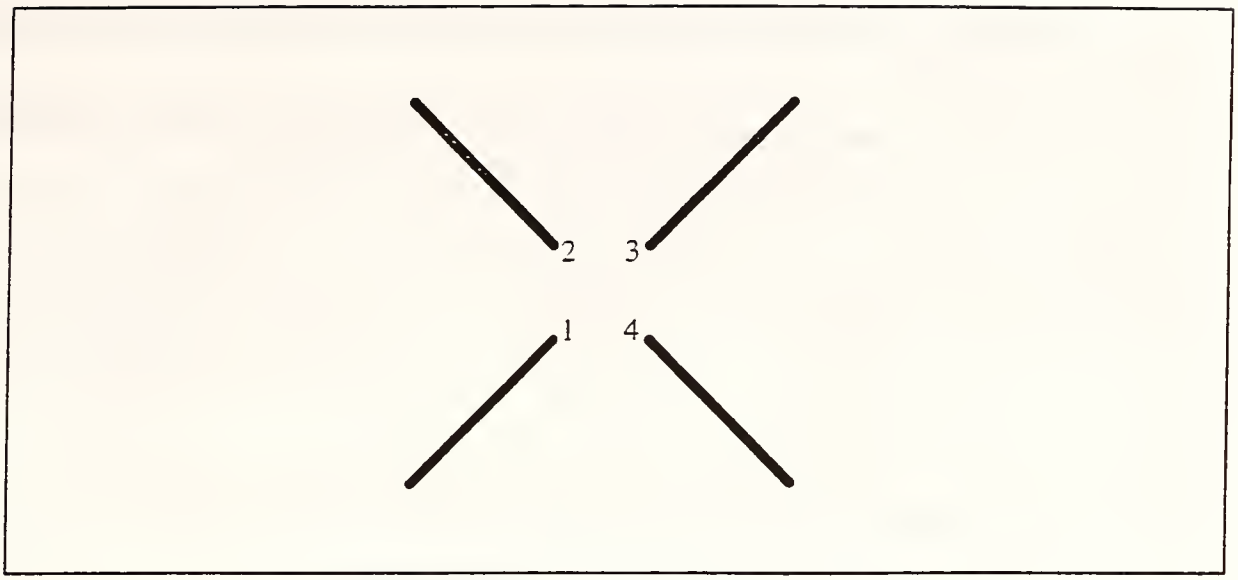


Figure 5.7 Detail of RC-8 Fiducial Mark

Two operators measured the images in the upper stage of a Kern DSR-1 in monocomparator mode. All measurements were in raw stage coordinates in micrometers. Using negatives, the first operator measured eight images for the calibration block. No pass points were measured, These measurements were used to determine the camera parameters. Using diapositives, the second operator measured eighteen photos of the strip. A total of 366 image points were measured, consisting of control and pass points. Though nineteen images were in the strip, the first image was not made available for measurement. Both operators measured the fiducials in the manner described above; however, the second operator placed the images in the stage ninety degrees clockwise from the first operator, These measurements were used to perform the aerotriangulation.

Two QBASIC programs were written to prepare the two different raw stage coordinates into the same format. The first program converted the calibration stage measurements to millimeters, performed a six parameter transformation (two scales, two rotations, and two translations) for the stage irregularities , and performed a rotation to orient the way the negatives were placed on the stage. The second program performed the same operations with the strip stage measurements but without my rotation.

A C program was written to determine the fiducial centers for each photo. First the center of each fiducial was determined by taking the intersection of the lines between the measured points. Then the fiducial center was determined by the intersection of lines between the calculated center of each fiducial. The fiducials were rotated around the fiducial center to form a 45 degree angle between fiducial number 1, the fiducial center, and fiducial number 3. The stage coordinates were translated to an image coordinate system based on the coordinates of the fiducial coordinates of the image.

Another C program was written, to statistically analyze the fiducial coordinates. The following tables show the results.

Fiducial	mean (mm)		σ (μm)	
	x	y	x	y
1	-106.088	-106.088	12.276	12.276
2	-106.028	106.092	12.533	10.260
3	106.070	106.067	10.730	10.730
4	106.036	-106.100	11.344	16.151

Table 5.1 Average Fiducial Coordinates for Calibration Photos

Fiducial	mean (mm)		σ (μm)	
	x	y	x	y
1	-106.050	-106.050	8.770	8.770
2	-106.012	106.051	12.048	8.324
3	106.040	106.040	9.037	9.037
4	106.011	-106.050	13.880	16.151

Table 5.2 Average Fiducial Coordinates for Strip Photos

The results show the average fiducial coordinates for the strip photos were better than the calibration photos. It was decided that the average fiducial coordinates for the strip photos would be used as the calibrated fiducial coordinates. The image coordinates were reduced to the common calibrated fiducial coordinates.

An additional note should be made about the measurement of the pass points. An attempt was made to measure easily identifiable points in the images without pre-marking the photos. Unfortunately, the photos consisted of farm fields. Some of the chosen pass points in one photo were just plow marks and became unidentifiable in the third photo of the triple overlap.

Shutter Delay

The circuitry for the photo diode and the trigger timing were wire wrapped onto a circuit board and bolted to a board with dimensions to cover the photo plate on the camera. The interface between the camera and the GPS receiver was connected to the circuit board. A light source was

placed underneath the camera to illuminate the photo diode when the shutter opened.

For each sample, the external trigger was grounded causing the ADC to be inhibited from sampling the data. The computer program was run to the point where the ADC was waiting for the data to be acquired. A button was pushed manually to send a signal to the shutter and to send +5 volts to the external trigger pin. Data was recorded in the cache until it was full. The cache was transferred to an array, which was then written to a file. Ten samples were taken at a rate of 2500 Hz, ten samples at a rate of 10,000 Hz, and one sample at a rate of 15,000 Hz. Each sample was written into separate file for further analysis. A schematic of the shutter timing setup is shown in Figure 5.8.

Figure 5.8 Shutter Timing Setup

CHAPTER 6
DATA ANALYSIS

A note should be made at this point about the coordinate system used throughout the remainder of this paper. The control point file provided by Dr. Merchant used MAD1 as the center of the local Cartesian system based on East, North, and Up directions. To ensure that the coordinates of the ground control within the ODOT calibration and test range were positive numbers, an offset of 10,000 meters was added to the East and North coordinates. The coordinates of MAD1 are (10000, 10000, 294.256). The benefits for using the local cartesian coordinate system come from not having to correct for earth curvature and not having to make coordinate transformations that may or may not add some ambiguity to the solution.

GPS Post-Processing

The data from each receiver was off-loaded to a computer. A static survey was performed to determine the location of the receivers occupying the ground points and the airplane while it was stationary over the orientation rose.

The time during which the airplane remained stationary produced an ambiguity solution with the dual-frequency receivers. The single-frequency receivers require longer periods of time before a solution may be found. Thus the kinematic method produces larger error values during the beginning of the solution. However, the solution may be determined either going forward in time, or going backward in time. A weighted mean was used to produce a solution with good error values throughout the flight. The solution was performed using Ashtech's PNAV software, version 2.

The results of the post-processing give the time (in GPS seconds since Sunday midnight UTC), and the east, north, and up values of the receiver in one second intervals. The east, north, and up values were centered on MAD1. The results begin at time 319583 seconds. At time 320507 seconds (924 seconds, or 15.4 minutes later), the results show the aircraft moving. At time 323435 seconds (2928 seconds, or 48.8 minutes later), the aircraft has stopped moving. The data ends at time 324645 seconds. A computation of the position of the orientation rose was performed for the airplane before takeoff and after landing. The difference in the two computations are dE = 30 mm, dN = -12 mm, and dU = 10 mm.

The results of the post-processing also give the times and positions of the events as recorded in the receiver. A number of the events were double events. The positions of the events were determined using linear interpolation.

Figure 6.1 shows the results of the survey. The flight path begins when the airplane began moving, and ends when the airplane stops moving. Also shown are the camera events as recorded with respect to the receiver and the location of three of the occupied ground stations.

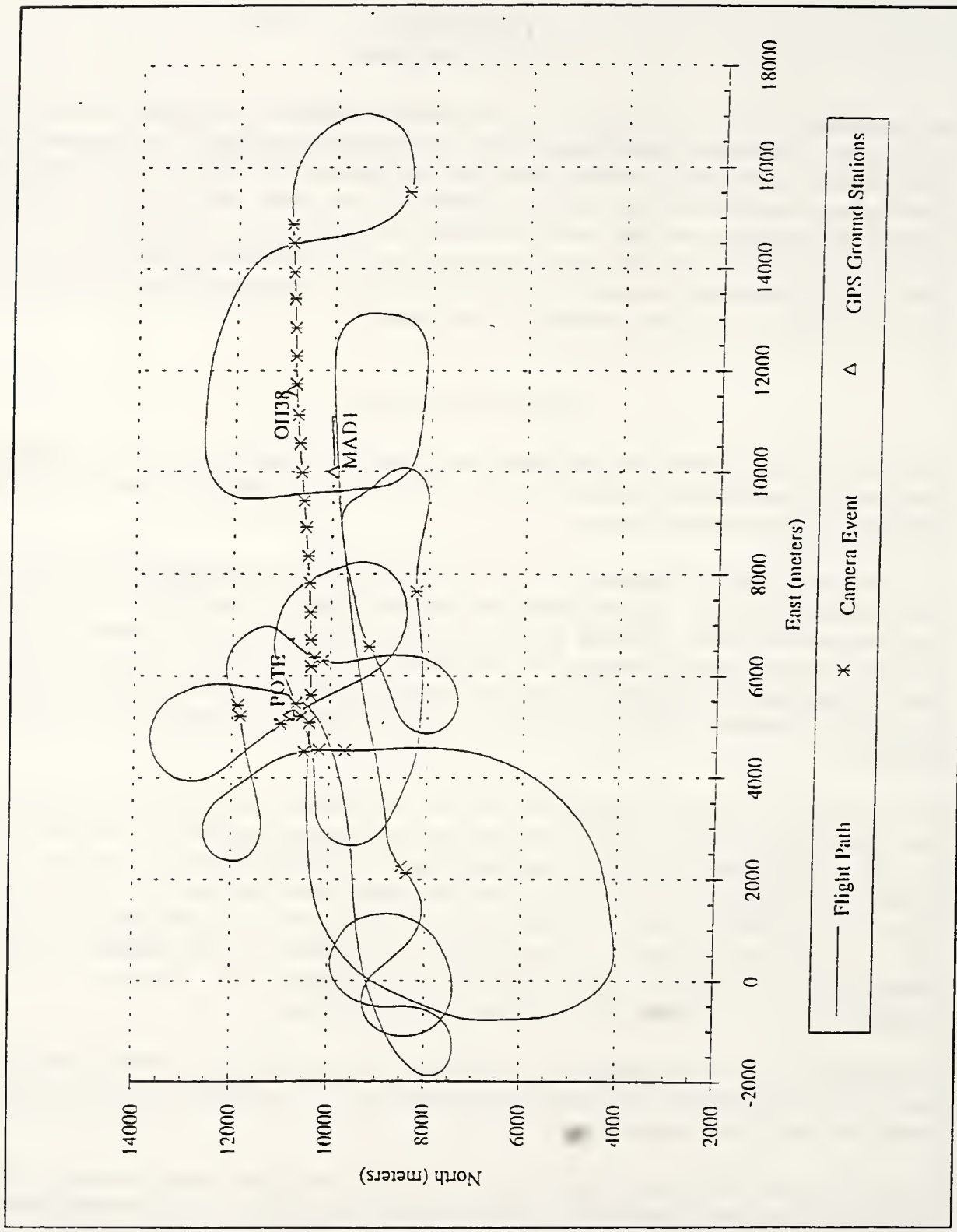


Figure 6.1 Flight Path with Shutter Events and Ground Stations

The theodolite measurements were used as observations in an adjustment program. The local coordinate system was then rotated and shifted to place the origin at the camera fiducial center, with the xy plane in the film plane and the x-axis along the airplane (straight and level flight) axis. The results of the program gave the x, y, z coordinates of the sixteen targets. The coordinates of the antenna phase center were determined using the coordinates of the 5 cm and 10 cm marks of the 15 cm ruler. The coordinates of the fiducial center were determined by taking the average of the four fiducial coordinates.

The following table gives the final target coordinates based on the initial local coordinate system. The coordinates for the fiducials are targets 12, 13, 14, and 15. The coordinates of the 15 cm ruler at the 5 cm and 10 cm ranks are, respectively, targets 10 and 11.

Table 6.1 Final Target Coordinates

Target No.	x	y	z
1	226.129	-18.606	-155.546
2	244.222	-5.983	-58.010
3	223.744	-2.625	2.053
4	151.084	84.350	-51.010
5	309.018	-181.665	-55.320
6	272.533	-53.236	-112.424
7	297.972	-56.270	-92.618
8	327.809	-49.281	-92.434
9	270.567	-25.559	-92.434
10	345.376	-99.178	46.199
11	345.429	-99.161	41.255
12	299.865	-5.496	-68.950
13	316.680	-5.062	-68.948
14	316.643	-21.649	-68.862
15	300.066	-22.058	-68.914
16	235.178	-12.283	-106.792

The next task was to determine the rotation angles such that when applied to the target coordinates, the resulting system would be aligned along the axis of the airplane, a "straight and level" airplane coordinate system. The mathematical method for determining the rotation angles is as follows :

$$x' = M(x - x_{pp})$$

where,

$$x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

are the target coordinates in the arbitrary, measuring coordinate system and,

$$x_{pp} = \begin{bmatrix} x_{pp} \\ y_{pp} \\ z_{pp} \end{bmatrix}$$

are the means of the four fiducial coordinates in the measuring system.

$$M = M_{\omega} M_{\phi} M_{\kappa}$$

is the rotation matrix defined so that the XY plane is coincident with the plane of the fiducial marks, and X is coincident with the x-axis defined by the fiducial marks. The camera was aligned so that it had zero tilts with respect to the camera mount and with a kappa of zero. This then defined the previously mentioned "straight and level" airplane coordinate system, with origin in the image plane at the principal point. The rotations were chosen such that the fiducial coordinates in the measuring system fulfilled the following conditions,

$$x_{12} = x_{14}$$
$$z_{12} = z_{14}$$
and
$$z_{13} = z_{15}$$

Then shifting the origin along the z-axis by the magnitude of the focal length to the rear nodal point of the lens, and by the internodal spacing to the front (entrance) nodal point of the lens we have the "straight and level" system with origin at the entrance node.

$$x'' = x' + \begin{bmatrix} 0 \\ 0 \\ 15.24 + 13.30 \end{bmatrix}$$

The eccentricity vector can then be read from the differences from the coordinates of target 11 minus 5 cm (the antenna to target offset). The rotation angles (in radians) were calculated and found to be κ=-O.7665, ϕ=-O.0038, ϖ=-O.0015. The resulting eccentricity vector then was found to be

$$x'' = \begin{bmatrix} 0.865 \\ 0.360 \\ 1.313 \end{bmatrix} \text{ meters}$$

Figure 6.2 illustrates the eccentricity vector.

Camera Calibration

Since the INDOT camera is not calibrated, the calibration parameters needed to be determined. These parameters determine the focal length, the principal point offsets, and radial lens distortion. To estimate these parameters, they may be added as unknowns in the bundle program. Such an adjustment is known as a bundle block adjustment with added parameters or self-calibration (Kraus, 1993). The results of the adjustment give a solution to the calibration parameters and to the camera parameters. The image points were corrected for the calibrated fiducials. These points were used in a block adjustment with the added parameters for focal length, principal point offsets, and three coefficients of radial lens distortion. The approximations for the camera positions were obtained by resection procedures. The total RMS for the image points was 0.0087mm, and the discrepancies at the control points were RMS-E = 0.039 meters, RMS-N = 0.033 meters, and RMS-Z = 0.058 meters. The results of the block adjustment gave the following values for the exterior orientation parameters and for the added parameters.

Table 6.2 Exterior Orientation Parameters for Calibration Block without Atmospheric Refraction Correction

	Angles (radians)			Positions (meters)		
Photo	omega	phi	kappa	East	North	Up
1009	0.00925	-0.01940	-1.04611	5044.79	11017.36	1949.92
1010	0.00502	-0.01427	-1.06799	5205.63	10765.85	1951.21
1011	0.03049	-0.04499	-2.86400	5471.68	10695.29	1955.76
1012	0.02906	-0.02915	3.41165	5196.79	10580.56	1946.01
1014	0.43662	0.07997	3.33474	6581.08	9224.35	1925.33
1016	0.13177	0.31195	1.30006	6359.97	10325.73	1923.95
1018	-0.36124	0.02737	3.19464	5200.25	11837.17	1925.49
1019	0.02885	-0.38561	-1.42198	4505.83	10539.22	1950.96

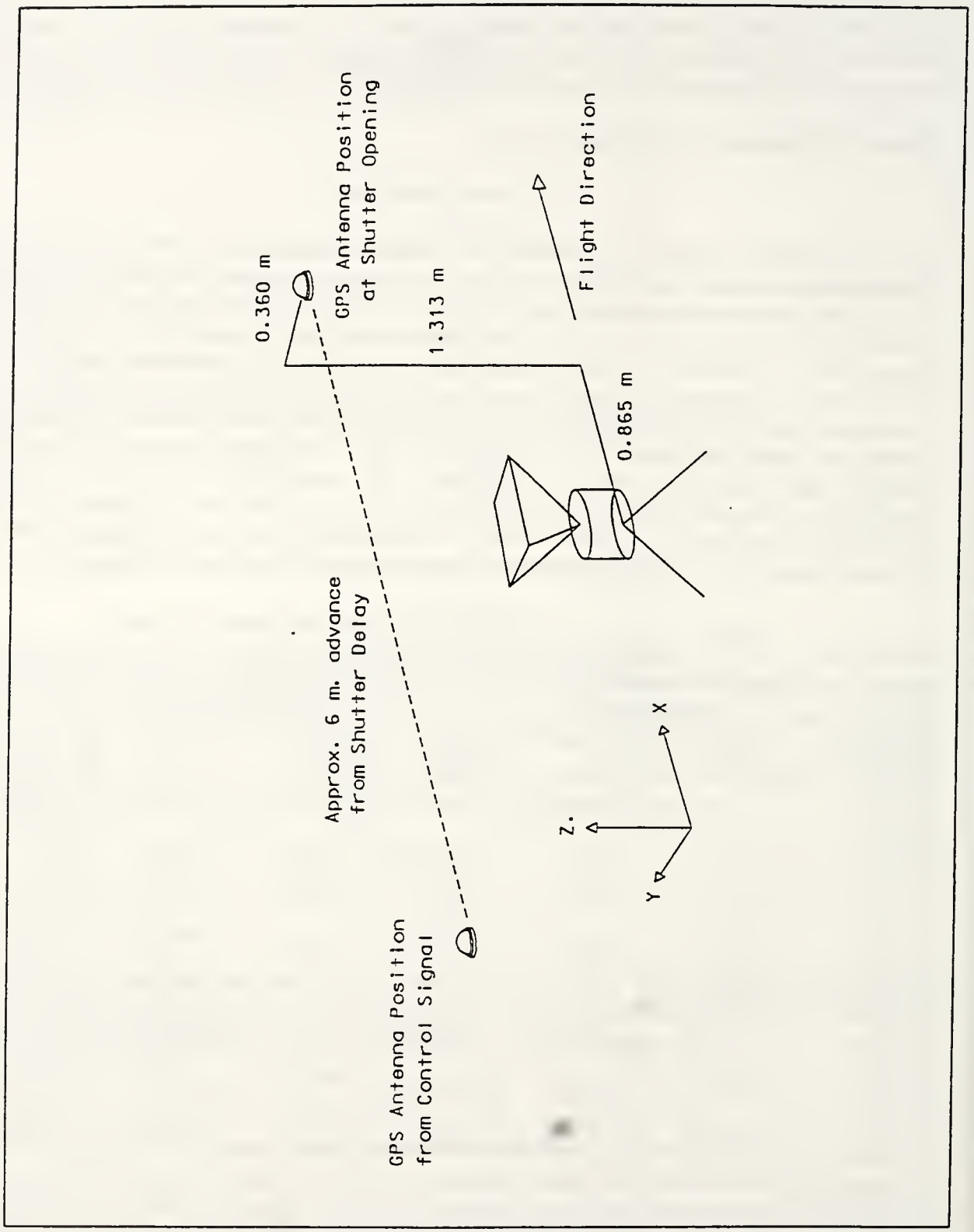


Figure 6.2 Eccentricity Vector

Table 6.3 Calibration Parameters for Solution without Atmospheric Refraction Correction

delta-f	0.094
delta-xo	-0.028
delta-yo	0.032
lens distortion, r^2	3.475E-03
lens distortion, r^4	-2.258E-04
lens distortion, r^6	3.077E-06

The fiducial coordinates were reduced to the origin of the estimated principal point and are tabulated below.

Table 6.4 Calibrated Fiducial Coordinates

Fiducial	x	y
1	-106.077	-106.018
2	-106.039	106.083
3	106.013	106.072
4	105.984	-106.018

The following figure gives the plot of the distortion curves. The nominal distortion curve for the RC-8 was found in Manual of Color Aerial Photography (ASPRS, 1968).

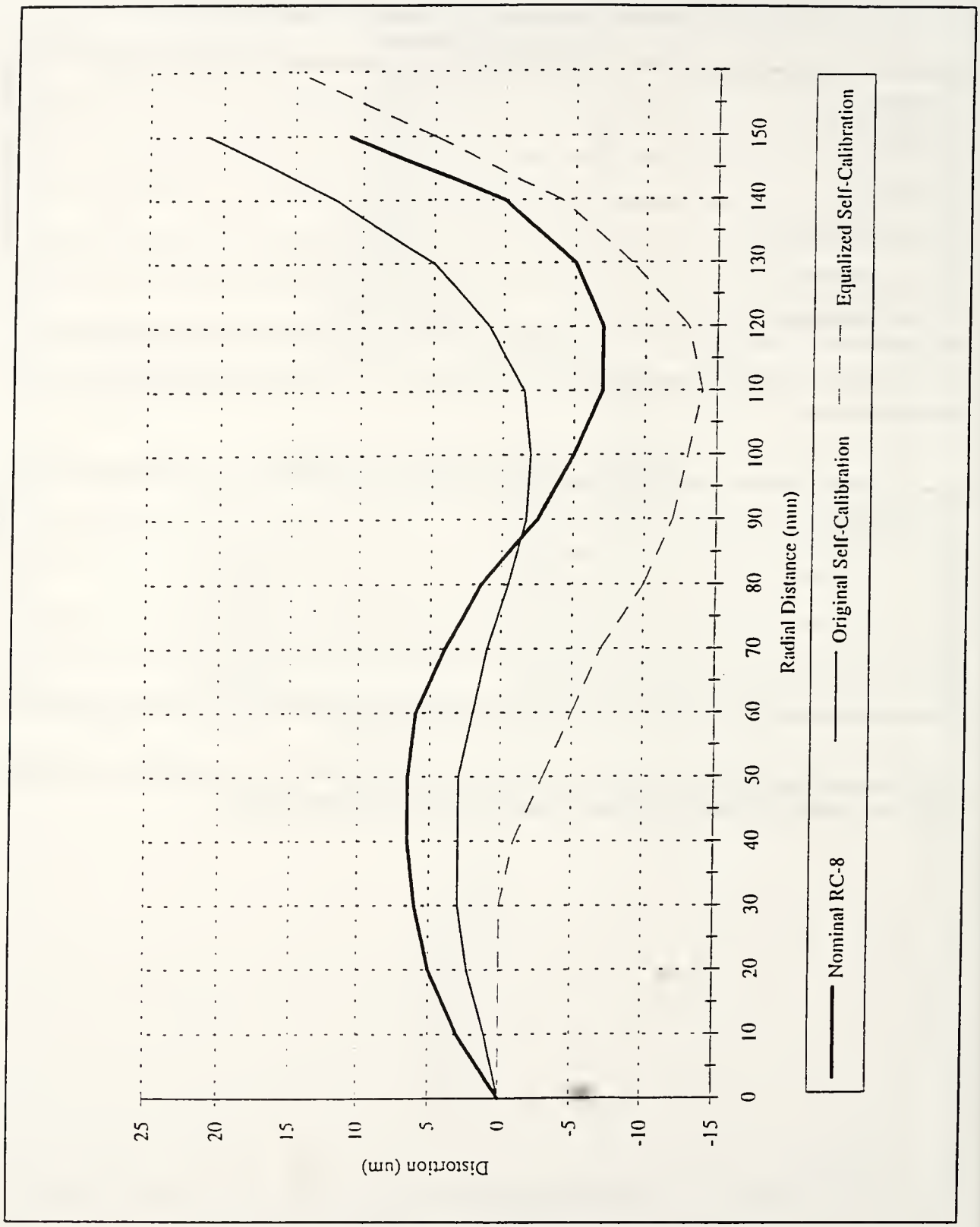


Figure 6.3 Lens Distortion Curves from Self-Calibration

After the image points were corrected for the radial lens distortion, the principal point offset, the calibrated focal length, it was discovered that one other correction needed to be performed. This correction is for atmospheric refraction. The available block adjustment program did not correct for the atmospheric refraction as an added parameter. The values for the calibrated focal length, the camera orientation angles and the flying height from the block adjustment with added parameters were used to correct for atmospheric refraction. All eight photos were rotated to vertical using the camera orientation angles. The correction for atmospheric refraction was then made to each of the measured image points using the ARDC (Air Research and Development Command of the US Air Force) Model Atmosphere (Moffitt and Mikhail, 1980.) The photo was then rotated back and scaled to the calibrated focal length. These image points, now corrected for atmospheric refraction, were used in a block adjustment with added parameters. The results of that adjustment we given in the following tables. The total RMS for the image points was 0.0087 mm, and the discrepancies at the control points were RMS-E = 0.040 meters, RMS-N = 0.032 meters, and the RMS-U = 0.058 meters. A comparison of the RMS values between the two block adjustments showed they were essentially the same. The results of the block adjustments gave the following parameters values.

Table 6.6 Calibration Parameters for Solution with Atmospheric Refraction Correction

	Angle (radians)			Position (meters)		
Photo	omega	phi	kappa	East	North	Up
1009	0.00925	-0.01940	-1.04611	5044.79	11017.37	1949.93
1010	0.00502	-0.01427	-1.06799	5205.63	10765.85	1951.22
1011	0.03049	0.04499	-2.86400	5471.69	10695.29	1955.78
1012	0.02906	-0.02915	3.41165	5196.79	10580.56	1946.03
1014	0.43664	0.07998	3.33473	6581.11	9224.32	1925.32
1016	0.13177	0.31198	1.30005	6360.00	10325.73	1923.94
1018	0.36126	0.02737	3.19464	5200.25	11837.20	1925.50
1019	0.02884	-0.38563	-1.42198	505.81	10539.23	1950.97

Table 6.3 Calibration Parameters for Solution with Atmospheric Refraction Correction

delta-f	0.095
delta-xo	-0.027
delta-yo	0.032
lens distortion, r^2	3.632E-03
lens distortion, r^4	-2.366E-04
lens distortion, r^6	3.442E-06

The results of the bundle adjustment with added parameters using image coordinates corrected for atmospheric refraction show no great difference from the results of the bundle adjustment with added parameters using image coordinates uncorrected for atmospheric refraction. These results were not used to refine the image coordinates. Considering the accuracy of the preliminary results from correcting for the shutter delay, the amount of time and effort to correct the image points using the results from bundle adjustment with added parameters corrected for atmospheric refraction was not justifiable. The effects for further correcting the image points for atmospheric refraction would be drowned out in the noise of the errors un-modeled in the shutter delay system.

Shutter Delay

A QBASIC program was written to analyze the data acquired from the ADC. The data files contained 4096 numbers with a possible range of 0 to 255. Each number corresponds to a voltage level that changed 39.1 millivolts per level. The position of the number in the data file corresponds to the time it was recorded. The change in time depends on the acquisition rate. For 2500 Hertz it represents a change of 0.4 ms, for 10,000 Hertz, 0.1 ms, and for 15,000 Hertz, 0.067 ms. An assumption was made that the acquisition of the data began at the same time as the signal to the shutter.

The program determines for each sample how many milliseconds to the pulse, the voltage level of the signal, and the length of the pulse. The shutter delay and duration times were analyzed with the aid of a Microsoft EXCEL program. It was found that the samples acquired at a rate of 2500 Hz are too variant. The samples acquired at a rate of 10,000 Hz provided adequate accuracy, but the 15,000 Hz was better. The following is a tabulation of the shutter time averages. In all cases, the duration lasted longer than 1/400 of a second (2.5 ms).

Table 6.7 - Average Shutter Delay Time (2500 Hertz)

2500 Hertz (10 samples)			
Time to signal (ms)		Duration (ms)	
Mean	93.52	Mean	3.12
Standard Deviation	27.54	Standard Deviation	0.16

Table 6.8 - Average Shutter Delay Time (10,000 Hertz)

10000 Hertz (10 samples)			
Time to signal (ms)		Duration (ms)	
Mean	129.57	Mean	2.79
Standard Deviation	25.18	Standard Deviation	0.03

Table 6.9 Average Shutter Delay Time (15,000 Hertz)

15000 Hertz (1 sample)			
Time to signal (ms)		Duration (ms)	
Recorded	140.20	Recorded	2.73

Table 6.10 Average Shutter Delay Time (All Frequencies)

All Frequencies (21 samples)			
Time to signal (ms)		Duration (ms)	
Mean	112.91	Mean	2.94
Standard Deviation	31.47	Standard Deviation	0.20

The shutter delay time was set at 0.113 s. Using the average velocity of the airplane during the acquisition of the strip, this delay equates to approximately 6 meters in distance. This shutter delay time will be used to correct for the time of the camera events.

The following figures represent the result of the samples.

Figure 6.4 is an overlay of the samples recorded at 2500 Hertz. Since the rate of 2500 Hertz only produced eight samples, and in one case seven samples of the curve, the curve is rather crudely defined.

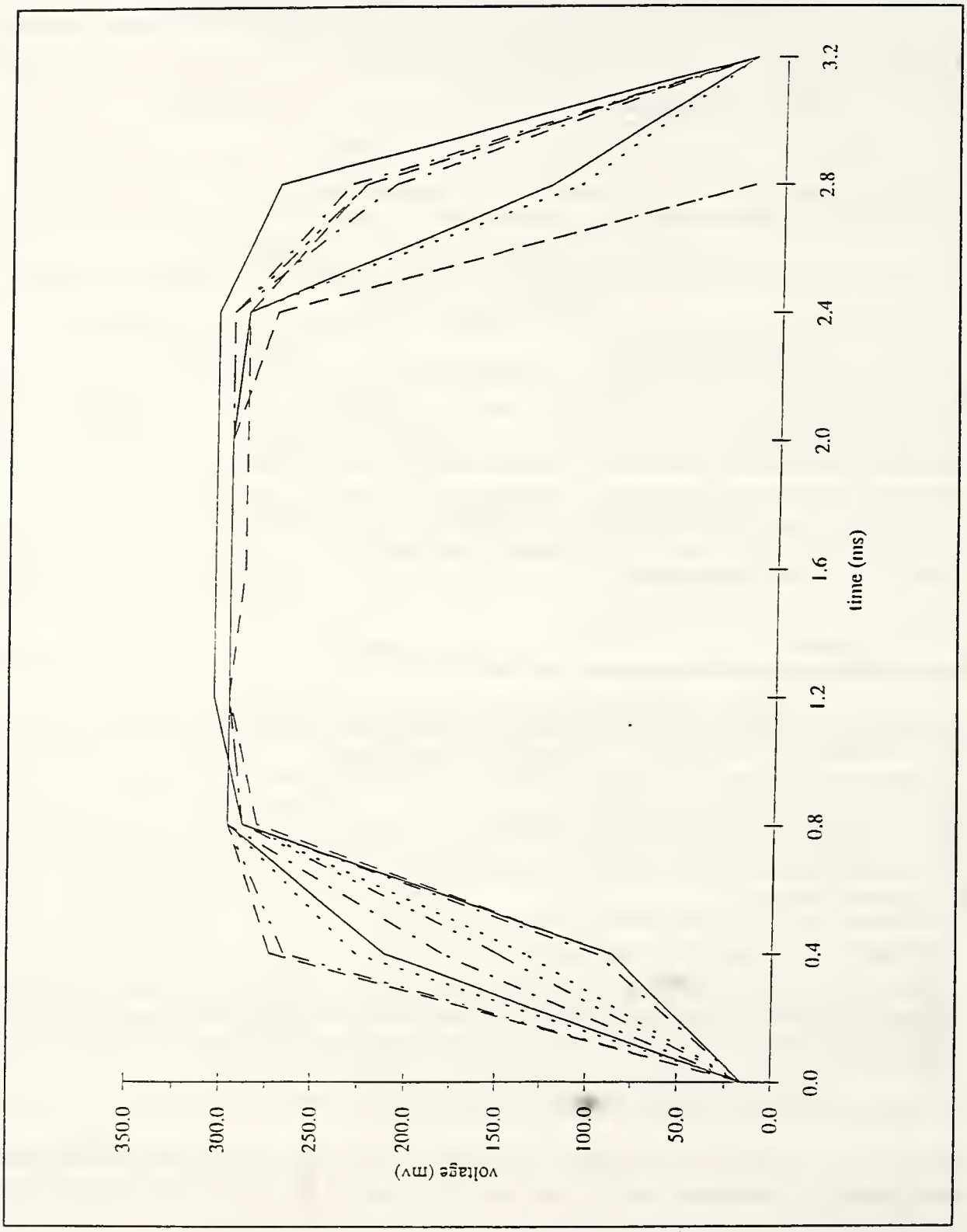


Figure 6.4 Overlay of ten 2500 Hz. Shutter Curves

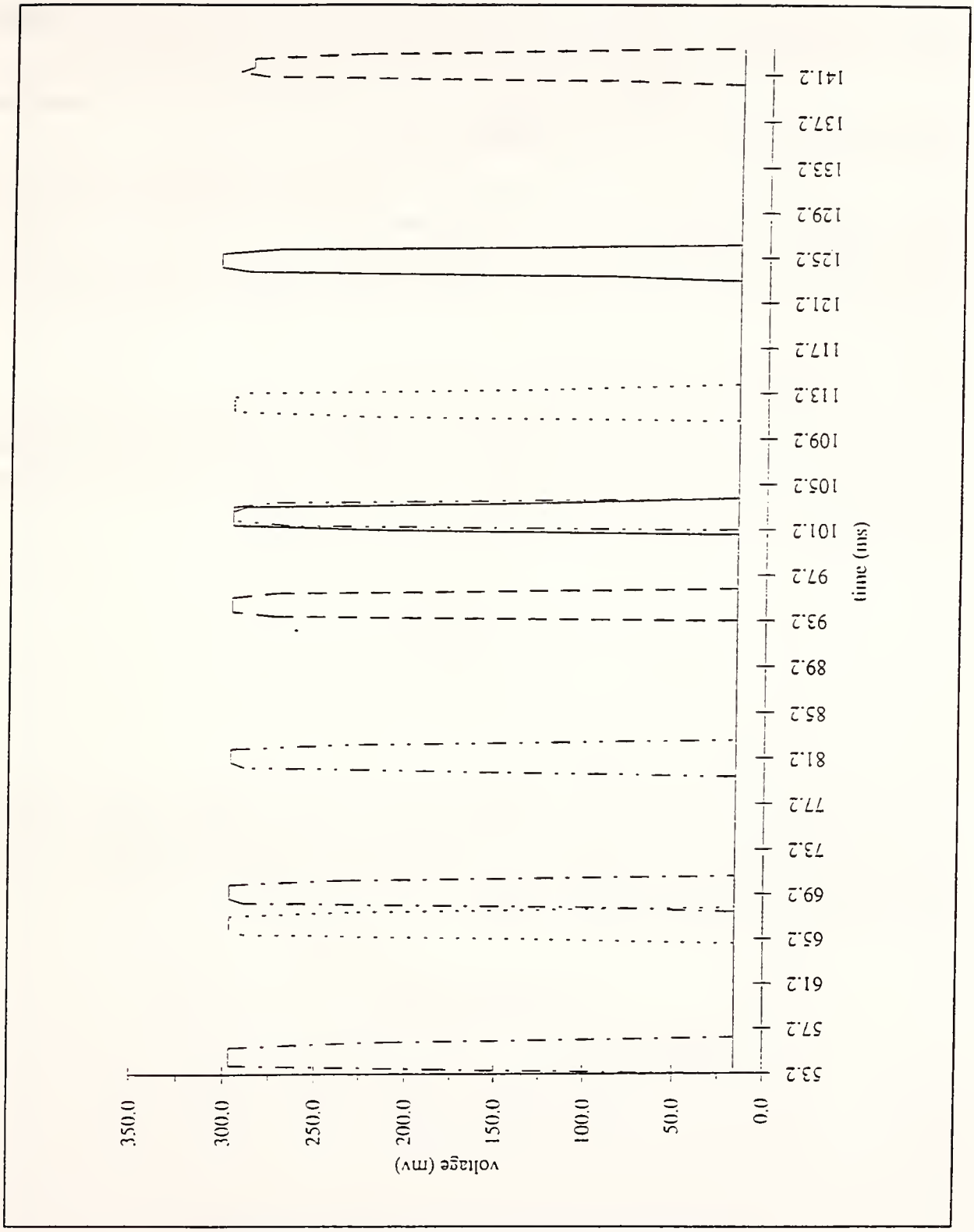


Figure 6.5 2500 Hz. Shutter Delay Distribution

Figure 6.5 represents the distribution of the samples recorded at 2500 Hertz. This distribution shows no cluster of delay times. This result is consistent with a uniform distribution.

Figure 6.6 represents an overlay of the samples recorded at 10,000 Hertz. At this sample rate, the curve of the exposure time is better defined.

Figure 6.7 represents the distribution of the samples recorded at 10,000 Hertz. Notice that the samples seem to be clustering with one of the samples appearing to be an outlier. This suggests that perhaps, the delay time is not uniform. However, when computed to the samples from the 2500 Hertz sample rate, the value appears more consistent with the others.

Figure 6.8 represents the sample recorded at 15,000 Hertz.

The efficiency of the shutter can be determined from Figure 6.8. If t_1, equals the time it takes for the shutter to completely open, t_2 equals the time when the shutter remains open for a time depending upon the exposure time, and t_3 equals the time it takes for the shutter to completely close, then the shutter efficiency can be expressed by the equations:

$$t_e = \frac{t_1 + t_3}{2} + t_2,$$

$$t_0 = \textit{total time}$$

efficiency of the shutter is the ratio t_e/t_o.

In this particular case, using the values obtained from the sample obtained at a 15,000 Hertz, $t_1 = t_3 = 0.6$ ms, and $t_2 = 1.6$ ms. This gives the value of t_e to be 2.2 ms. The total time $t_o = 2.8$ ms, which gives the value for the efficiency of the shutter to be the ration 2.2 ms/2.8 ms = 78.6%. The efficiency of the rotary shutter for the RC-8 at full aperture of *f*/5.6 is stated to be 83% (American Society of Photogrammetry, 1968).

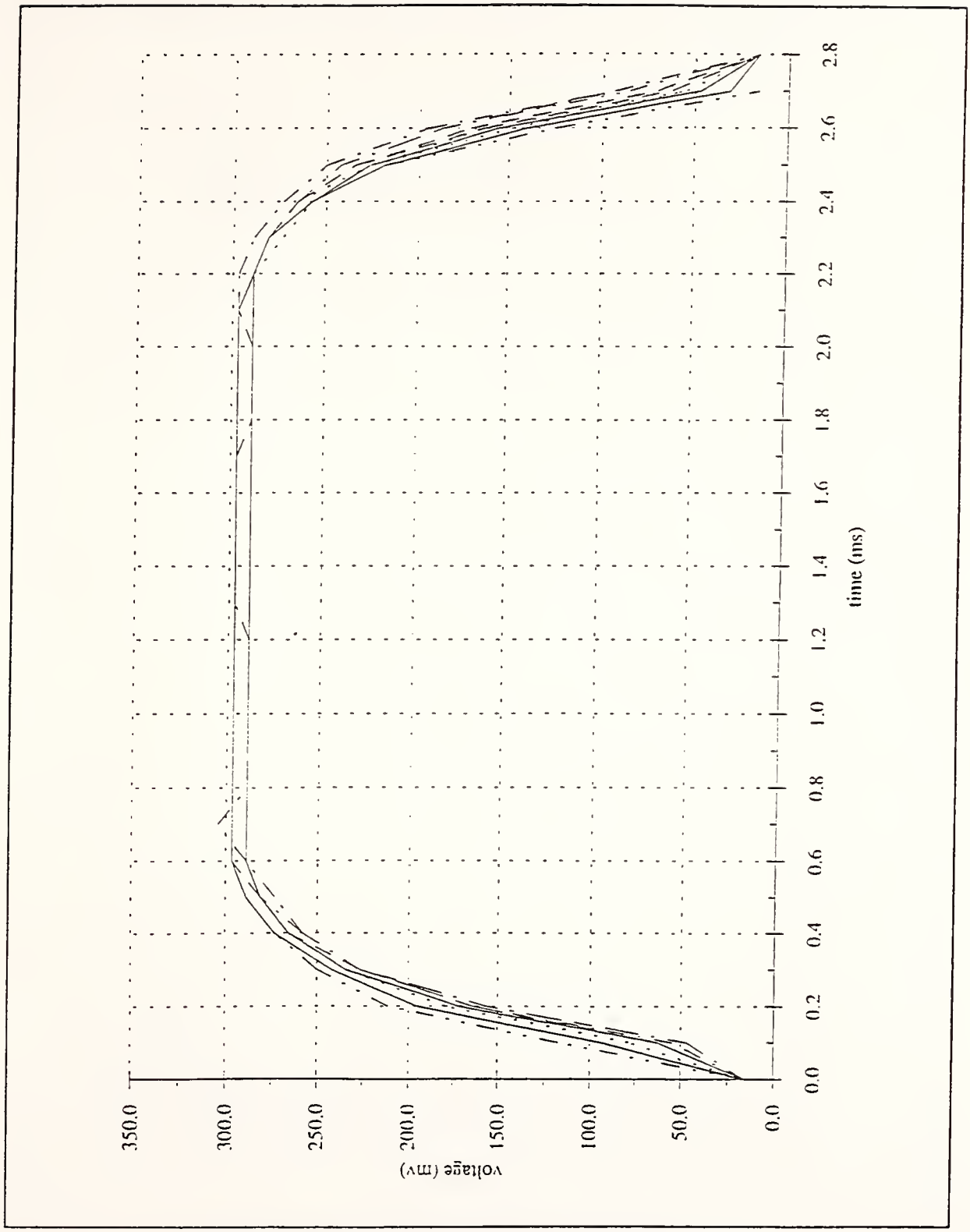


Figure 6.6 Overlay of Ten 10,000 Hz. Shutter Curves

Figure 6.7 10,000 Hz. Shutter Delay Distribution

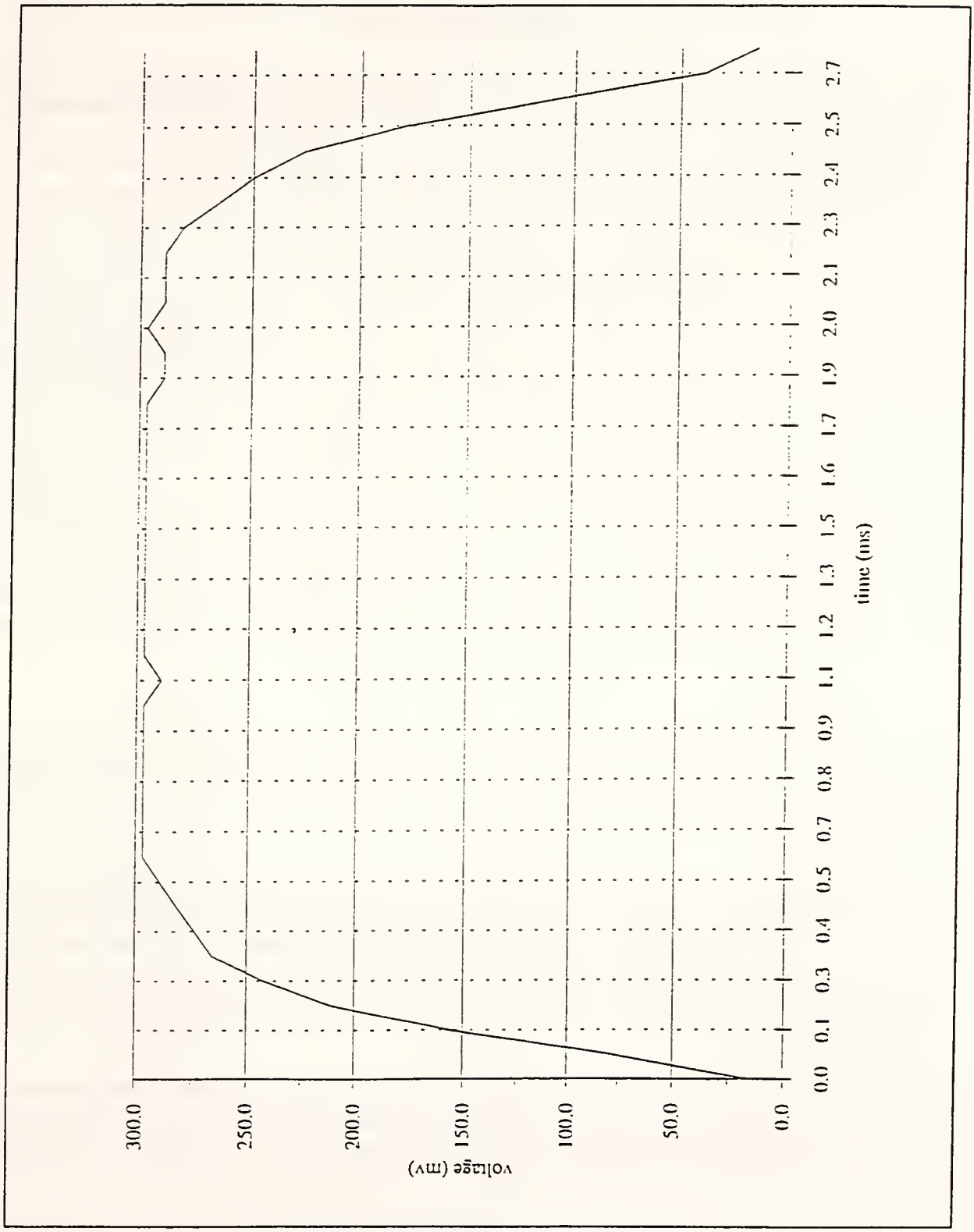


Figure 6.8 15,000 Hz. Shutter Curve

Image Measurement

Since the observations to the block program are the image points, a quick summary of how these observations have been refined is in order. First the image points were measured in a monocomparator mode. These measurements were made in micrometers in the stage coordinate system. These raw stage coordinates were transformed to correct for film deformations and for the non-orthogonality of the stage arms and converted from micrometers into millimeters. The centers of the fiducial marks and the fiducial center for each image were found. The image measurements were transformed from the stage coordinate system into an image coordinate system based on the individual fiducial center of the image. A statistical study of the fiducial coordinates determined the calibrated fiducial coordinates. The image coordinates for each image were reduced to a common calibrated fiducial system. These coordinates were used as observations to a block adjustment with added parameters. From this block adjustment with added parameters, the radial lens distortion curve, the principal point offset, and the equivalent focal length were found. A radial lens correction curve and calibrated focal length were found. The image coordinates were reduced further to the principal point offset and corrected for the radial lens distortion. Also, rotation parameters and camera heights from this block adjustment with added parameters were needed to correct for atmospheric refraction. These atmospheric corrections were made to the same image coordinates that were used in the block adjustment with added parameters. The block adjustment with added parameters was computed again, however no corrections to the image points were made as a result of correcting for atmospheric refraction. The charts in Figures 6.9 and 6.10 illustrate the various corrections made to the image points.

Block Adjustment without GPS

The analysis of the image points is broken into two phases. First the block adjustment is made using none of the GPS derived camera positions. The solution of this block adjustment is used as a reference to compare the accuracy of the block adjustment using the GPS derived camera positions.

The block adjustment was first run with all the image points in the solution. The sigma values for the control points were 1.0×10^{-5} meters, while the sigma values for the image points were 0.005 mm. The RMS values for this solution was rather large; and so, a process was begun to weed out the "bad" observations. The last photo of the strip, Photo 1040, has no control points. One of the pass points had to be dropped due to a large residual; and the photo was dropped altogether.

The following is a table of the various images, the number of image points, and a summary of the type of image point eliminated during the weeding out process. The total RMS for the image points after eliminating "bad" image points was 0.0118 mm. The RMS for the 96 control points for East, North, and Up was 0.046, 0.069, and 0.112 meters respectively.

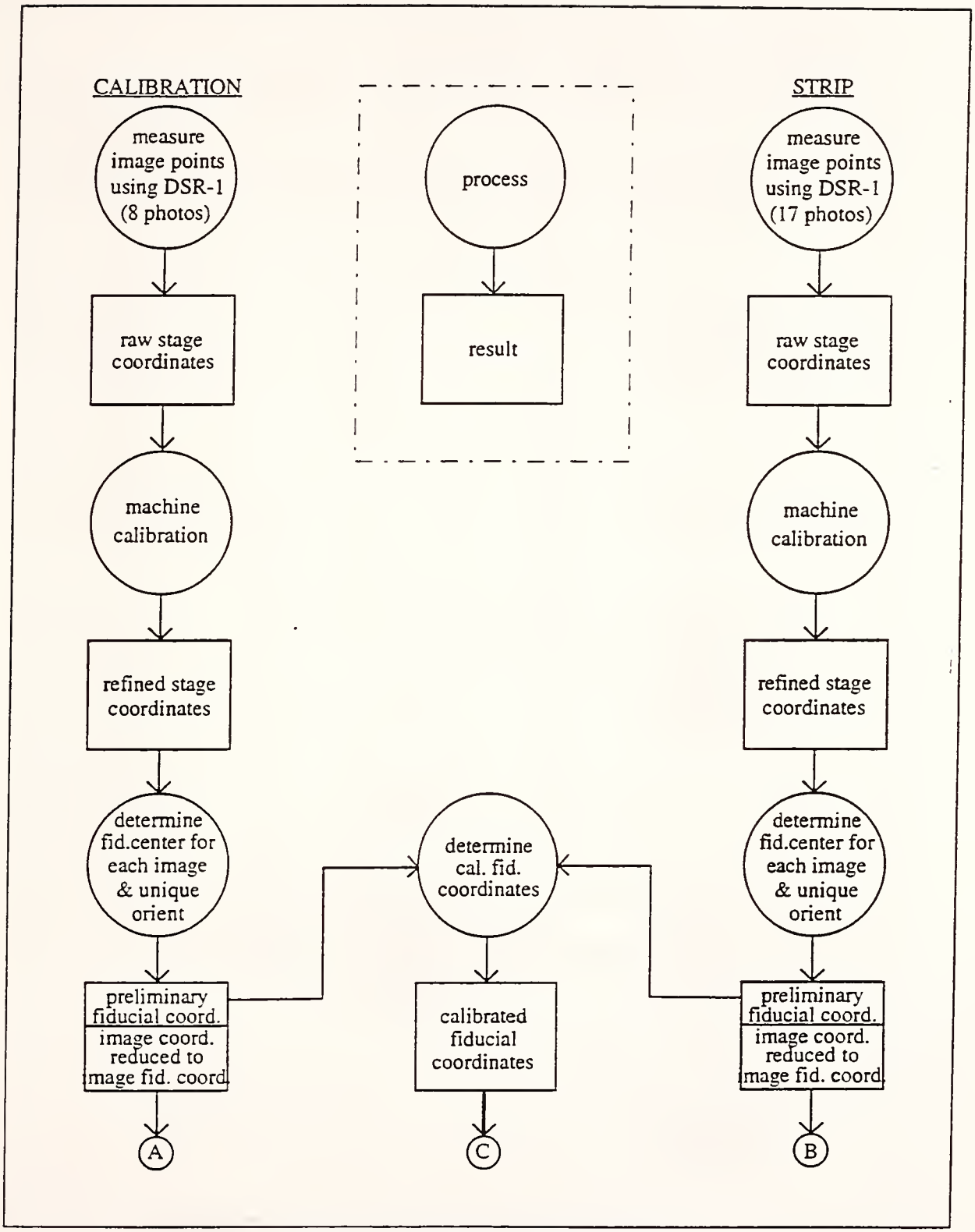


Figure 6.9 Reduction of Image Points

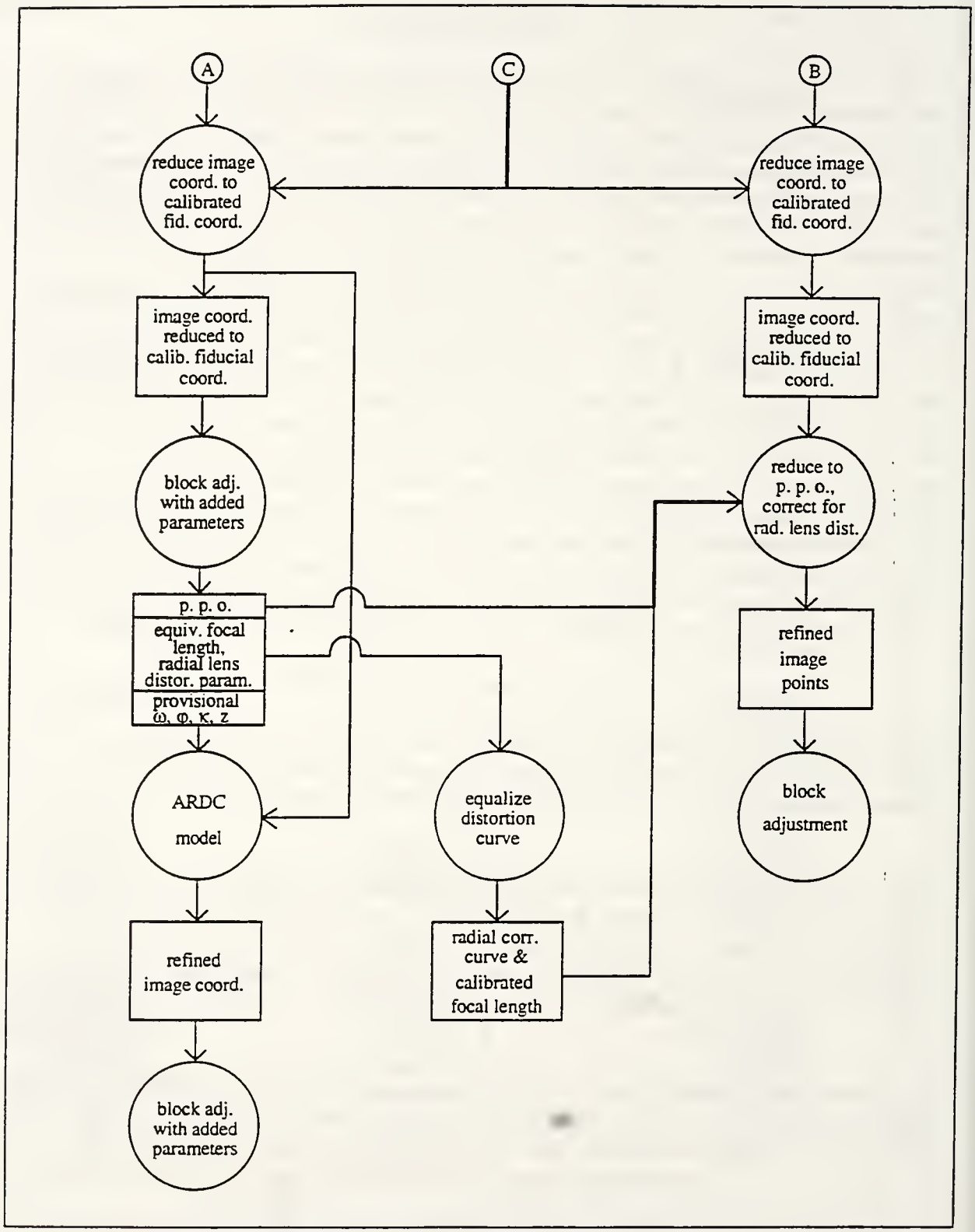


Figure 6.10 Reduction of Image Points (continued)

Image Number	Number of Image Points	Total Number of Points Eliminated	Number of Pass Points Eliminated	Number of Control Points Eliminated
1023	20	1	0	1
1024	24	0	0	0
1025	12	1	0	1
1026	12	2	1	1
1027	13	1	0	1
1028	16	3	1	2
1029	23	5	1	5
1030	25	4	3	1
1031	28	2	1	2
1032	27	2	1	1
1033	31	2	1	1
1034	29	1	1	0
1035	28	1	0	1
1036	26	2	2	0
1037	21	1	0	1
1038	15	2	2	0
1039	10	0	0	0
Totals	360	30	14	16

Table 6.11 Distribution of Edited Image Points

The final solution for the camera parameters are tabulated below.

Table 6.12 - Final Station Parameters

Photo number	ω	ϕ	κ	East (m)	North (m)	Up (m)
1023	-0.017389	-0.012585	0.089945	5621.217	10385.793	1337.976
1024	-0.038254	0.001720	0.049754	6170.988	10385.545	1339.543
1025	-0.032983	-0.011966	0.025926	6713.895	10396.094	1349.264
1026	0.000597	0.018137	0.039695	7255.167	10415.333	1356.942
1027	-0.018767	0.002829	0.052165	7804.809	10439.012	1351.933
1028	-0.018605	-0.010317	0.101544	8352.644	10480.495	1355.172
1029	-0.028279	-0.035766	0.050665	8906.897	10527.391	1371.497
1030	-0.002920	-0.004297	0.083810	9441.093	10576.313	1381.821
1031	-0.043765	0.015415	0.021004	9981.629	10626.687	1376.331
1032	-0.052446	0.032713	0.011996	10546.970	10679.271	1363.097
1033	0.006033	0.019179	-0.114678	11124.392	10725.973	1349.445
1034	-0.024336	0.006699	0.044637	11709.664	10758.699	1346.781
1035	0.003202	-0.022553	0.121313	12285.978	10779.885	1357.841
1036	-0.013142	-0.006996	0.152235	12844.414	10795.015	1361.090
1037	-0.011484	0.008117	0.085167	13388.924	10816.902	1355.486
1038	-0.045059	0.011768	0.066475	13941.342	10840.418	1351.785
1039	0.013355	-0.013247	0.067228	14487.909	10870.267	1356.006

Block Adjustment with GPS

The GPS post processing adjustment program (PNAV) produced the coordinates of the GPS antenna at the times of the recorded events. These positions are centered at MAD1, and were easily transformed to the east-north-up system.

The next step was to determine what time/event corresponded to which photograph. This could be done by matching the positions produced from the GPS program to the positions obtained from the bundle adjustment. For instance, Photo 1023 was matched to the time of 322627.317 seconds (event number 2). Consequently, Photo 1024 could be matched up with time/event number 3. Photo 1025 could not be matched with time/event number 4; however, it could be matched up with time/event number 5. This was an example of the double event produced by the reed relay. The matching up of the times with the photos was fairly straight forward except for Photo 1027. Photo 1026 was matched with event number 6. The time difference between event number 6 and event number 7 was 8.985. The average time between camera events was 8.597 seconds, while the average time when a double event occurred was 0.387 seconds. The sum of the two averages is 8.984 seconds. Thus recording of event number 7 was the double event without recording the prior actual event. The time for event for Photo 1027 was set to be 8.597 seconds after Photo 1026, or at 322661.668 seconds. The next event was 8.588 seconds later, which falls in line with the other differences.

The following table gives the times and positions of the antenna for the strip. The positions were transformed into the local coordinate system. The differences in time between events are also tabulated. When a double event occurs, the differences between two times are shown. The events used for the comparison are marked. The * in the "Used" column shows the location of the time difference for Photo 1027.

Table 6.13 - GPS Solution for the
Antenna Position at Recorded Event

Event Number	Time (sec)	East (m)	North (m)	Up (m)	Time Diff.	Time Diff.	Used
1	322619.172	5078.387	10401.201	1341.353			
2	322627.317	5616.813	10384.373	1340.827	8.145		X
3	322635.882	6165.474	10384.497	1341.910	8.565		X
4	322636.270	6190.062	10384.987	1342.032	0.388		
5	322644.472	6708.273	10395.643	1351.462	8.202	8.590	X
6	322653.071	7249.888	10414.531	1359.100	8.599		X
7	322662.056	7825.469	10439.565	1353.872	8.985		*
8	322670.256	8349.951	10479.505	1356.618	8.200		X
9	322678.853	8897.428	10525.658	1372.475	8.597		X
10	322679.237	8921.704	10527.690	1373.557	0.384		
11	322687.448	9434.618	10575.433	1383.071	8.211	8.595	X
12	322696.046	9978.626	10625.879	1377.721	8.598		X
13	322704.634	10539.044	10678.085	1364.719	8.589		X
14	322705.021	10564.684	10680.522	1363.932	0.386		
15	322713.248	11118.754	10724.743	1351.071	8.227	8.614	X
16	322721.852	11703.806	10757.679	1348.339	8.604		X
17	322722.240	11730.120	10758.892	1348.665	0.387		
18	322730.454	12280.971	10779.115	1359.358	8.214	8.601	X
19	322739.043	12838.485	10793.753	1363.179	8.590		X
20	322747.650	13384.582	10815.725	1357.877	8.606		X
21	322748.035	13408.973	10816.818	1357.463	0.385		
22	322756.257	13932.388	10839.182	1354.547	8.222	8.607	X
23	322764.858	14483.537	10868.769	1358.729	8.601		X
24	322765.244	14508.184	10870.236	1359.177	0.386		
25	322771.012	14876.013	10891.865	1373.733	5.768	6.154	

These events were recorded at the time when the signal from the intervalometer was sent to the receiver. The times for when the camera opened the shutter occurred 0.113 seconds later. Thus each time was advanced by that amount. The following table shows the times when the exposures for each photo occurred.

Table 6.14 - Exposure Times

photo	time	photo	time
1023	322627.430	1032	322704.747
1024	322635.995	1033	322713.361
1025	322644.585	1034	322721.965
1026	322653.184	1035	322730.567
1027	322661.781	1036	322739.156
1028	322670.369	1037	322747.763
1029	322678.966	1038	322756.370
1030	322687.561	1039	322764.971
1031	322696.159		

The position of the camera at these times was interpolated from the GPS solution data. This data contained the East, North, Up positions of the receiver and had been thinned out to 1.0 seconds. First, the Lagrange interpolation method was tested by using the original times of the events and computing the differences given by the PNAV software. These differences show a zero mean and varied by(σ_E=0.016, σ_N=0.012, and σ_U = 0.033 meters.

Thus Lagrange interpolation was used to determine the event positions. The positions were translated into the local system and the eccentricity vector was added. The differences between the bundle adjustment and the GPS positions was then calculated. The r.m.s. values for the differences were 1.977 meters for east, 0.503 meters for north and 0.795 meters for up. The following tables summarizes the results. Table 6.15 shows the differences from the interpolated positions and the positions obtained from the block adjustment as given in Table 6.12. Table 6.16 gives the statistics for the differences.

Table 6.15 - Interpolated Positions and
Differences from Positions Obtained from Block Adjustment

		Interpolated Positions (m)			Differences (m)		
Photo	time	East	North	Up	dE	dN	dU
1023	322627.430	5623.302	10384.607	1339.438	2.085	-1.186	1.462
1024	322635.995	6171.776	10384.975	1340.686	0.788	-0.570	1.143
1025	322644.585	6714.528	10396.131	1350.187	0.633	0.037	0.923
1026	322653.184	7256.164	10415.102	1357.592	0.997	-0.231	0.650
1027	322661.781	7806.881	10438.813	1352.258	2.072	-0.199	0.325
1028	322670.369	8356.269	10480.473	1355.311	3.625	-0.022	0.139
1029	322678.966	8903.733	0526.609	1371.521	-3.164	-0.782	0.024
1030	322687.561	9440.809	10576.507	1381.923	-0.284	0.194	0.102
1031	322696.159	9985.018	10626.877	1376.354	3.389	0.190	0.023
1032	322704.747	10545.647	10679.140	1363.164	-1.323	-0.131	0.067
1033	322713.361	11125.533	10725.651	1349.597	1.141	-0.322	0.152
1034	322721.965	11710.617	10758.393	1347.120	0.953	-0.306	0.339
1035	322730.567	12287.581	10779.618	1358.235	1.603	-0.267	0.394
1036	322739.156	12844.847	10794.399	1361.789	0.433	-0.616	0.699
1037	322747.763	13390.859	10816.385	1356.410	1.935	-0.517	0.924
1038	322756.370	13938.753	10839.826	1353.159	-2.589	-0.592	1.374
1039	322764.971	14489.889	10869.556	1357.546	1.980	-0.711	1.540

Table 6.16 - Statistics for Differences Between Camera Positions

	East (m)	North (m)	Up (m)
Mean	0.283	0.113	0.123
RMS	1.977	0.503	0.795
σ	1.880	0.499	0.616

Various block adjustment scenarios were then investigated using the interpolated positions. The intent here was to investigate how the accuracy of the block adjustment changed depending on where the control points were located in the strip and the number of control points used.

In each investigation the same number of image points were used, and in fact, used the same image points in the bundle adjustment without GPS. The same six control points were used in various combinations. These six control points were picked to fall in the triple overlap area and in the corners and middle of the strip. The following figure shows the location and name of the control points along with MAD 1 and the interpolated camera events for reference.

The block adjustment program computes the positions of the rest of the control points as check points. These check points can then be subtracted from the control points and r.m.s. calculated to test the accuracy of the procedure.

Using GPS derived camera stations as control with ground control was compared with using only ground control. The sigma values for the camera stations in the first case was set . 2 meters. Six scenarios were run, and in each case the number of ground control was reduced one. The sigma values for the camera stations in the second case was set at 10,000 meters. n a similar manner, four scenarios were run reducing each case by one ground contro point. e following two tables give the results of the investigation. The accuracies are r. s. vai and are in meters. It should be noted that using no ground control points with th GPS d ived camera stations will produce a divergent bundle solution. Also, using less than tw ground ontrol points will produce a divergent bundle solution.

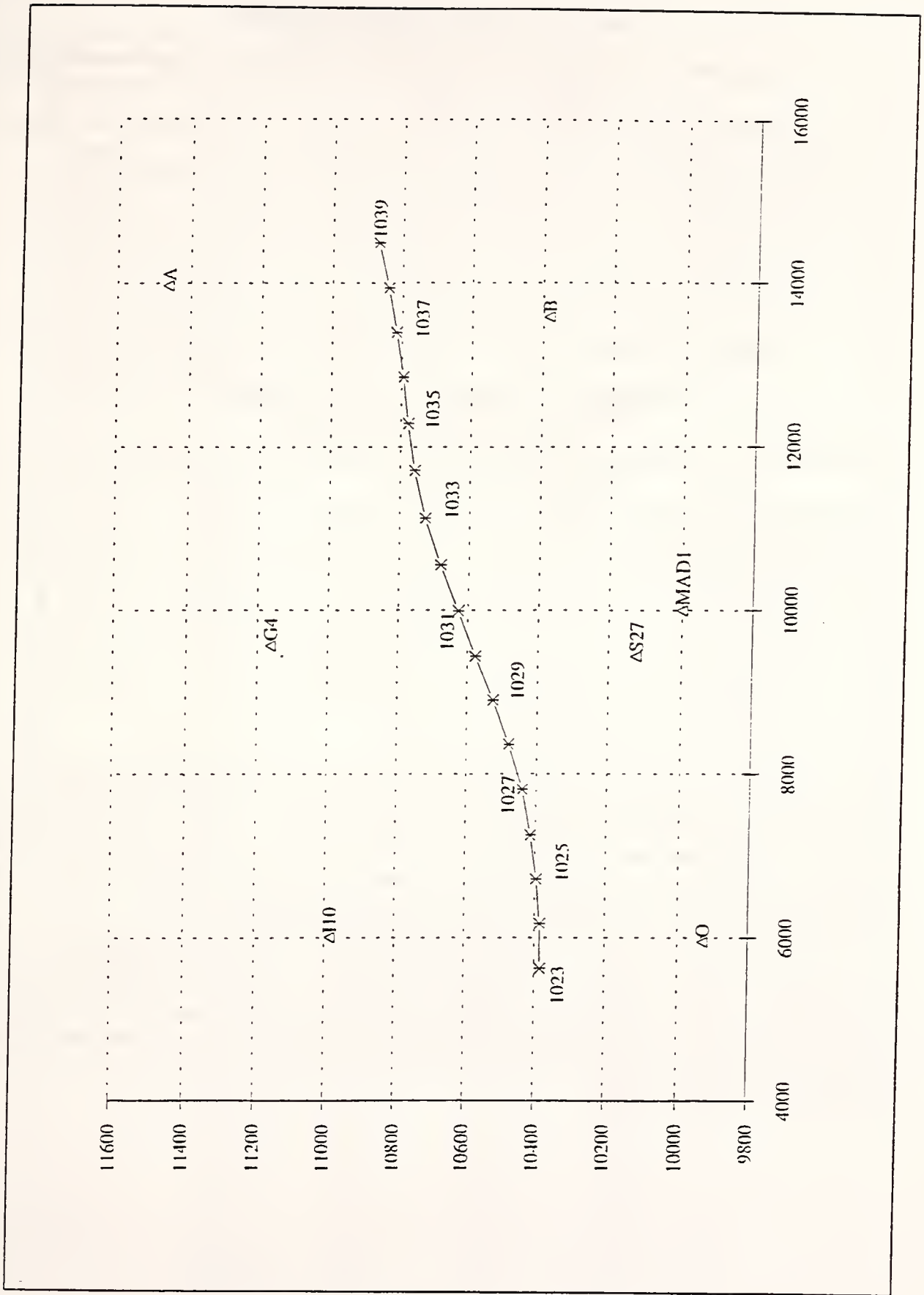


Figure 6.11 Exposure Stations and Control Points

Table 6.17 - Bundle Adjustment Accuracy
using GPS Derived Camera Stations

Control Points Used	# Control Points	# Check Points	East	North	Up
O+A+I10+B+G4+S27	6	91	0.136	0.120	0.406
O+A+I10+B+G4	5	92	0.195	0.191	0.411
O+A+I10+B	4	93	0.377	0.212	0.902
O+A+I10	3	94	0.154	0.504	0.839
O+A	2	95	1.019	1.128	1.040
O	1	96	0.850	0.806	0.993

Table 6.18 - Bundle Adjustment Accuracy
using Ground Control Points Only

Control Points Used	# Control Points	# Check Points	East	North	Up
O+A+I10+B+G4+S27	6	91	0.135	0.121	0.413
O+A+I10+B+G4	5	92	0.197	0.191	0.426
O+A+I10+B	4	93	0.422	0.194	1.447
O+A+I10	3	94	0.166	0.487	1.261
O+A	2	95	No Convergence		
O	1	96			

Three separate investigations were conducted to examine the accuracy of the GPS derived camera stations in various locations along the strip. In the first part, six bundle adjustments were run using only one ground control point. Ninety-six check points were used to test the accuracy of each run. In the second part, nine bundle adjustments were ran using two ground control points, the only criteria in choosing which two was that they must straddle the axis of the strip. Ninety-five check points were used to test the accuracy of each run. In the third part, six bundle adjustments were run using three ground control points. The criteria for choosing which three control points to use was that two of the control points were on the extremes of the strip and the third control point was across the axis. The following tables give the results of this investigation. The RMS values in the last column are for each point (or points), while the RMS values in the last row are for each coordinate. The r. m. s. values in the lower right corner represent the overall accuracy of the results.

Table 6.19 - Single Control Point Accuracies

Control Point Used	East	North	Up	r.m.s.
O	0.850	0.806	0.993	0.887
A	1.102	1.349	1.116	1.194
I10	0.677	1.011	0.907	0.876
B	0.852	0.990	0.848	0.899
G4	0.906	1.076	1.073	1.021
S27	0.549	1.020	0.696	0.780
RMS	0.841	1.102	0.958	0.973

Table 6.20 - Two Control Point Accuracies

Control Points Used	East	North	Up	r.m.s.
O+A	1.019	1.1285	1.040	1.063
O+I10	0.501	1.174	0.756	0.857
O+G4	0.288	1.305	0.892	0.928
A+B	1.118	1.285	1.042	1.153
A+S27	0.544	1.510	0.621	0.994
I10+B	0.648	1.186	0.954	0.955
I10+S27	0.635	1.251	0.743	0.917
B+G4	0.871	1.357	0.692	1.013
G4+S27	0.659	1.016	0.552	0.768
RMS	0.740	1.253	0.828	0.967

Table 6.21 - Three Control Point Accuracies

Control Points Used	East	North	Up	r.m.s.
O+A+I10	0.154	0.504	0.839	0.572
O+B+I10	0.114	0.339	0.851	0.533
A+B+I10	0.812	0.515	1.190	0.883
A+B+O	1.043	0.413	1.243	0.967
B+G4+O	0.163	0.184	0.476	0.309
A+I10+S27	0.155	0.270	0.647	0.415
RMS	0.553	0.390	0.916	0.657

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

It is appropriate here to remark on the types of errors and problems which were encountered. Some of the problems were experiential, such as having a nonfunctional GPS receiver or neglecting to record the crab angle of the airplane during the flight. One can only work around such problems, and try to avoid them in the future. Other problems were encountered only after analyzing the results. These problems can be solved for the next project, or at least, recommendations cm be made. In the following sections, some of these problems are explained and, recommendations are made.

Image Measurement

Problems were encountered with the measurement of the image points, particularly the pass points. The area of the mission was near corn fields, and considering the location of any future INDOT project, this may be a consistent encounter. It was very difficult to find well-defined image points in an unplowed corn field. Any future project should consider pugging the pass points, that is, drilling a small hole in the photograph at the location of the pass point. Also, the image points should be measured in stereo, since this will help differentiate the surrounding features.

Camera Calibration

When the bundle adjustment for added parameters was run using image points corrected for atmospheric refraction, the results showed little difference from the results when the bundle adjustment for added parameters was run using image points uncorrected for atmospheric refraction. When the correction for the atmospheric refraction was applied to the image points, the maximum correction was noted for each photograph. For the vertical photographs, the largest change was only on the order of 3 microns. For the oblique photographs, the largest change was 8 microns with the average change for one photo being 4 microns. With such changes in the image points, there should have been a greater change in the solution. This result indicates that the camera calibration parameters absorbed a large part of the atmospheric effect.

It was recommended that INDOT send their camera to be calibrated by the USGS laboratory. If they should actually do that, then it would be interesting to compare the results of the calibration from USGS and the bundle adjustment with added parameters.

Shutter Timing

The time from the recording of the event in the receiver to when the shutter opened produced perhaps the greatest mount of unmodeled error in the entire system. One can see this quite readily by observing the amount of variance in the shutter delay times. For the shutter time using all the samples, the average shutter time was 0.113 s with standard deviation of 31 ms. This

variance is too large to be able to accurately predict the opening of the shutter time since 31 ms is equivalent to 1.7 meters, assuming an aircraft moving at a ground speed of 200 km/hr (124 mph). Another problem which did not help the situation was the behavior of the reed relay in the interface between the intervalometer and the GPS receiver. It was thought that the second event recorded due to the reed relay could be ignored. This was not the case as can be seen with the investigation into time for Photo 1027.

Trying to determine the correct times after the fact, is not as good as using the actual time of the camera event. Therefore, it is recommended that the development or purchase of a system to determine the camera event at time of occurrence. This system would consist of a photo diode inside the lens cone of the camera. When light enters the cone, a signal would be sent from a non-mechanical component to the GPS receiver. Care should be taken to minimize the response time of the photo diode, while at the same time maximizing the sensitivity of the photo diode. There will be less light during an actual mission than an incandescent light as was used in the data acquisition of the shutter delay.

Bundle Adjustment

The differences between the bundle adjustment derived camera positions and the GPS derived camera positions are due to unmodeled effects. Atmospheric refraction was not modeled in the preprocessing of the image coordinates. The eccentricity vector was not modeled iteratively in the bundle adjustment. Unfortunately, due to an oversight, the crab angle (the difference between the "course" and "heading") was not recorded at the time of flight. This oversight prevented us from correctly modeling the absolute orientation of the eccentricity vector. By necessity, it was assumed that the "course" was identical to the "heading", which would be the case for zero crosswind. Weather data for the flight date is still being sought to try to enhance the estimate of crosswind effects.

A note should also be made about the correction for drift errors as per Ackermann and Schade, 1993. The bundle adjustment did not model the drift errors as suggested in the literature. No modeling of the drift errors is necessary since the post-processing techniques should correct for any existing drift errors. In particular, the use of the forward solution and the backward solution together to get a weighted solution should be very effective in correcting for drift errors.

Due to the unmodeled effects present in the bundle adjustment, the assessment of the results is difficult. The results of the investigation show little difference between the accuracy of the check points for the GPS derived camera stations as control points, and using only ground control. The use of the GPS derived camera stations as control points should be more accurate since, in effect, there are seventeen more control points being used. What should not be overlooked is the fact that results are available using only one or two control points. For the conventional method this produces a divergent solution to the block adjustment. One result is rather puzzling. The results for both cases show better results going from four ground control points to three control points in the east and up directions.

The interest in investigating how one, two and three control points changed the accuracy was for two reasons; first, to see how the accuracy changes with the various patterns of control. The results show a variation in the results from the different patterns. This variation may be due to the accuracy of the measured image points or to the position of control points in the strip. More investigation using simulated data is needed to determine which is the case. The second reason for this investigation was to see the overall effect of accuracy with the number of control points by getting a greater population of results. As expected the results show greater accuracy as the number of control points increases.

LIST OF REFERENCES

American Society of Photogrammetry, (1968), "Manual of Color Aerial Photogrammetry", First Edition, Falls Church, Virginia

American Society of Photogrammetry, (1980), "Manual of Photogrammetry", Fourth Edition, Falls Church, Virginia

Ackermann, F., (1992a), "Operational Rules and Accuracy Models for GPS- Aerotriangulation", International Archives of Photogrammetry and Remote Sensing, Commission III, Vol. XXIX, Part B3, pp. 691-700

Ackermann, F., (1992b), "Kinematic GPS Control for Photogrammetry", Photogrammetric Record, 14(80): pp. 261-276

Ackermann, F. and Schade, H., (1993), "Application of GPS for Aerial Triangulation", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. I 1, pp. 1625-1632

Bethel, J., Johnson, S., and van Gelder, B., (1993), "Use of GPS to Enhance Mapping by Photogrammetry", A Proposal for Joint Highway Research Project in cooperation with Indiana Department of Transportation and Federal Highway Administration

Bethel, J., Johnson, S., and van Gelder, B., (1995), "Accuracy of GPS Kinematic Vectors During Aerial Camera Calibration Flights", Paper presented during the session on "Geodetic Considerations in Airborne GPS-Assisted Photogrammetry" at the ACSM/ASPRS Spring Convention, Charlotte, NC, February 27-March 3, 1995

Bethel, J., Johnson, S., and van Gelder, B., (1995), "Use of GPS to Enhance Mapping by Photogrammetry", Final Report for Joint Highway Research Project No. HPR-2092

Burnside, C., (1985), "Mapping from Aerial Photographs", Second Edition, Wiley, New York

Colomina, I, (1993), "A Note on the Analytics of Aerial Triangulation with GPS Control", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. I 1, pp. 1619-1624

Crowl, J. and Merchant D., (1995), "Airborne GPS Photogrammetric Calibration and Test Ranges at Madison County, Ohio", 1995 ACSM/ASPRS Annual Convention and Exposition Technical Papers, Volume III, pp 603-612

Curry, S. and Schuckman, K., (1993), "Practical Considerations for the Use of Airborne GPS for Photogrammetry", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. 11, pp. 1611-1617

Friess, P, and Heuchel, T., (1992), "Experience with GPS-Supported Aerial Triangulation", International Archives of Photogrammetry and Remote Sensing, Commission I, Vol. XXIX, Part B1, pp. 299-305

Gruen, A., Cocard, M., and Kahle, H.-G., (1993), "Photogrammetry and Kinematic GPS: Results of a High Accuracy Test", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. 11, pp. 1643-1650

Habib, A. and Novak, K., (1994), "GPS Controlled Aerial Triangulation of Single Flight Lines", 1994 ASPRS/ACSM Annual Convention Technical Papers, Vol. I:ASPRS, Reno, Nevada

Jacobsen, K., (1992), "Handling of Disturbed Kinematic GPS-Data in Block Adjustment", International Archives of Photogrammetry and Remote Sensing, Commission 1, Vol. XXIX, Part B 1, pp. 306-311

Jacobsen, K., (1993), "Experiences in GPS Photogrammetry", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. 11, November 1993, pp. 1651-1658

Kraus, K., (1993), "Photogrammetry", Fourth Edition, Ferd. Dummlers Verlag, Bonn

Lapine, L., (1990), "Analytical Calibration of the Airborne Photogrammetric System Using a priori Knowledge of the Exposure Station Obtained from Kinematic Global Positioning System Techniques", Dept. of Geodetic Science and Surveying, The Ohio State University, UMI Dissertation Services, Publication No. 9111738, Ann Arbor, MI 48109

Merchant, D., (1993), "GPS Controlled Aerial Photogrammetry", Photogrammetric Engineering & Remote Sensing, Vol. 59, No. 11, pp. 1633-1636

Moffitt,F., Mikhail,E., (1980),"Photogrammetry",Third Edition, Harper & Row Publishers Inc., New York

Silicon Alley, Inc., (1989), "Etude 25MHz 8-bit Analog to-Digital Converter User's Guide", Silicon Alley, Inc.

Schuckman, K, Curry, S, Zhao, M, and Salsig, G., (1992), "A Practical Test of a Photogrammetric Project Controlled with Airborne GPS", International Archives of Photogrammetry and Remote Sensing, Commission I, Vol. XXIX, Part B 1, pp. 312-316

Seeber, G., (1993), "Satellite Geodesy",Walter de Gruyter, Berlin

Torge, W., (1991), "Geodesy", 2nd. Ed., Walter de Gruyter, Berlin

APPENDIX

Computer programs developed for the project including independent model block adjustment with supporting procedures and subroutines, suitable for data acquired from analog stereoplotters such as the INDOT B8's.

```
/* indmod.c - 17-apr-95 */
/* simultaneous independent model block adjustment */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "mat.c"
#include "gauss.c"

#define MAXPOINTS 200
#define MAXMODS 20
#define MAXOBS 500
#define MAXITER 5

struct point
{
char point_name[16];
double x,y,z;
int control_flag[4];
} points[MAXPOINTS];

struct model
{
int model_name;
double omega,phi,kappa;
double scale;
double tx,ty,tz;
} models[MAXMODS];

struct ob
{
char point_name[16];
int model_name;
int logical_model_number;
double xm,ym,zm;
} obs[MAXOBS];

int get_logical_model_number(int mn, int allow_add);
int get_logical_point_number(char *s, int allow_add);

int nmod,npnt,nobs;

main (int arg, char *argv[])
{
int i,j,k;
FILE *in_read,*out_write,*in_appx,*in_cont;
char ch,dumstr[128],point_id[16];
int itmp,jtmp;
int allow_add;
double omega,phi,kappa,tx,ty,tz,scale;
int model_n;
double x,y,z;
int lmn,test,ip,io;
int iter,terminated;
double avg_dx,avg_dy,avg_dz;
int avg_count;
double **m,**mm,**mw,**mp,**mk,*v1,*v2,*dx;
double **ndot,*tdot;
double **nddot[MAXPOINTS],**nbar[MAXPOINTS],*tddot[MAXPOINTS];
double **bdot,**bddot,*f;
double **ndot_add,*tdot_add,**nddot_add,*tddot_add,**nbar_add;
double *delta_dot,*delta_ddot;

printf("open files\n");
if ((in_read=fopen("indmod.inp", "rt")) == NULL)
```

```
  {
  printf ("error opening indmod.inp");
  exit (1) ;
  }

if ((in_appx=fopen("indmod.apx", "rt")) == NULL)
  {
  printf ("error opening indmod.apx");
  exit (1) ;
  }

if ((in_cont=fopen("indmod.con", "rt")) == NULL)
  {
  printf ("error opening indmod.con");
  exit (1) ;
  }

if ((out_write=fopen("indmod.out", "wt")) == NULL)
  {
  printf ("error opening indmod.out");
  exit (1) ;
  }

printf("read observations\n");

/* read observations and initialize observation and model arrays */

nmod=0;
npnt=0;
nobs=0;
while( !feof(in_read))
  {
  nobs++;
  if(nobs >= MAXOBS)
    {
    printf("too many observations\n");
    exit(1);
    }
  fscanf (in_read, "%d %16s %lf %lf %lf%[ \n]", &obs[nobs].model_name,
    &obs[nobs].point_name,&obs[nobs].xm,&obs[nobs].ym,&obs[nobs].zm,dumstr);
  allow_add=1; /* true */
  itmp=get_logical_model_number(obs[nobs].model_name,allow_add);
  allow_add=1; /* true */
  jtmp=get_logical_point_number(obs[nobs].point_name,allow_add);
  }

printf("read model initial approximations\n");

/* read in initial approximations for model transformation parameters */

while( !feof(in_appx))
  {
  fscanf (in_appx, "%d %lf %lf %lf %lf %lf %lf %lf %[ \n]",
          &model_n,&omega,&phi,&kappa,&tx,&ty,&tz,&scale,dumstr);
  for(i=1; i<=nmod; i++)
    {
    if(models[i].model_name == model_n)
      {
      models[i].omega=omega;
      models[i].phi=phi;
      models[i].kappa=kappa;
      models[i].tx=tx;
      models[i].ty=ty;
      models[i].tz=tz;
      models[i].scale=scale;
```

```
      }
    }
  }
fclose(in_appx);

printf("put logical model number in observation record\n");

/* fill in logical model number in the observation record */

for(i=1; i<=nmod; i++)
  {
  for(j=1; j<=nobs; j++)
    {
    if(obs[j].model_name == models[i].model_name)
          obs[j].logical_model_number=i;
    }
  }

printf("read control points\n");

/* read in control points */

for(i=1; i<=npnt; i++)
  {
  points[i].control_flag[1]=0;
  points[i].control_flag[2]=0;
  points[i].control_flag[3]=0;
  }

while( !feof(in_cont))
  {
  fscanf (in_cont, "%16s %lf %lf %lf %[ \n]",point_id,&x,&y,&z,dumstr);
  for(i=1; i<=npnt; i++)
    {
    if(strcmp(points[i].point_name,point_id) == 0)
      {
      points[i].x=x;
      if(x != 0.0) points[i].control_flag[1]=1;
      points[i].y=y;
      if(y != 0.0) points[i].control_flag[2]=1;
      points[i].z=z;
      if(z != 0.0) points[i].control_flag[3]=1;
      }
    }
  }
fclose(in_cont);

printf("compute point initial approximations\n");

/* compute initial approximations for the points */

m  = mat_alloc (3,3);
v1 = vec_alloc (3);
v2 = vec_alloc (3);
mm = mat_alloc (3,3);
mw = mat_alloc (3,3);
mp = mat_alloc (3,3);
mk = mat_alloc (3,3);
dx = vec_alloc (3);

for(i=1; i<=3; i++)
  for(j=1; j<=3; j++) mw[i][j]=0.0;
mw[2][3]=1.0;
mw[3][2]=-1.0;
```

```
for(i=1; i<=3; i++)
  for(j=1; j<=3; j++) mk[i][j]=0.0;
mk[1][2]=1.0;
mk[2][1]=-1.0;

for(i=1; i<=3; i++)
  for(j=1; j<=3; j++) mp[i][j]=0.0;

for(i=1; i<=npnt; i++)
  {
  /* search for an observation with this point */
  test=1;
  j=0;
  while((test != 0) && (j <= (nobs-1)))
    {
    j++;
    test=strcmp(obs[j].point_name,points[i].point_name);
    }
  if(test != 0)
    {
    printf("no match in point approx comps\n");
    exit(1);
    }
  lmn=obs[j].logical_model_number;
  omega=models[lmn].omega;
  phi=models[lmn].phi;
  kappa=models[lmn].kappa;
  tx=models[lmn].tx;
  ty=models[lmn].ty;
  tz=models[lmn].tz;
  scale=models[lmn].scale;
  ROTM(m,omega,phi,kappa);
  v1[1]=m[1][1]*obs[j].xm + m[2][1]*obs[j].ym + m[3][1]*obs[j].zm;
  v1[2]=m[1][2]*obs[j].xm + m[2][2]*obs[j].ym + m[3][2]*obs[j].zm;
  v1[3]=m[1][3]*obs[j].xm + m[2][3]*obs[j].ym + m[3][3]*obs[j].zm;
  x=tx + (1.0/scale)*v1[1];
  y=ty + (1.0/scale)*v1[2];
  z=tz + (1.0/scale)*v1[3];
  if(points[i].control_flag[1] != 1) points[i].x=x;
  if(points[i].control_flag[2] != 1) points[i].y=y;
  if(points[i].control_flag[3] != 1) points[i].z=z;
  }

printf("allocate memory\n");

/* allocate memory */

ndot=mat_alloc(nmod*7,nmod*7);
ndot_add=mat_alloc(nmod*7,nmod*7);
tdot=vec_alloc(nmod*7);
tdot_add=vec_alloc(nmod*7);
nddot_add=mat_alloc(3,3);
tddot_add=vec_alloc(3);
nbar_add=mat_alloc(nmod*7,3);
delta_dot=vec_alloc(nmod*7);
delta_ddot=vec_alloc(3);

for(i=1; i<=npnt; i++)
  {
  nddot[i]=mat_alloc(3,3);
  tddot[i]=vec_alloc(3);
  nbar[i]=mat_alloc(nmod*7,3);
  }

bdot=mat_alloc(3,7*nmod);
```

```
bddot=mat_alloc(3,3);
f=vec_alloc(3);

printf("begin iterations\n");

/* iterate */

terminated=0;
iter=0;

while((terminated == 0) && (iter < MAXITER))
  {
  iter++;

  /* zero matrices */

  for(i=1; i<=7*nmod; i++)
    {
    for(j=1; j<=7*nmod; j++)
      {
      ndot[i][j]=0.0;
      }
    tdot[i]=0.0;
    }

  /* loop through the points */

  for(ip=1; ip<=npnt; ip++)
    {
    for(i=1; i<=3; i++)
      {
      for(j=1; j<=3; j++) nddot[ip][i][j]=0.0;
      }
    for(i=1; i<=7*nmod; i++)
      {
      for(j=1; j<=3; j++) nbar[ip][i][j]=0.0;
      }
    for(i=1; i<=3; i++) tddot[ip][i]=0.0;

    for(io=1; io<=nobs; io++)
      {
      test=strcmp(obs[io].point_name,points[ip].point_name);
      if(test == 0)
        {
        for(i=1; i<=3; i++)
          {
          for(j=1; j<=7*nmod; j++) bdot[i][j]=0.0;
          }
        lmn=obs[io].logical_model_number;
        omega=models[lmn].omega;
        phi=models[lmn].phi;
        kappa=models[lmn].kappa;
        tx=models[lmn].tx;
        ty=models[lmn].ty;
        tz=models[lmn].tz;
        scale=models[lmn].scale;
        ROTM(m,omega,phi,kappa);
        mp[1][2]=sin(omega);
        mp[1][3]=-cos(omega);
        mp[2][1]= -mp[1][2];
        mp[3][1]= -mp[1][3];
        dx[1]=points[ip].x - tx;
        dx[2]=points[ip].y - ty;
        dx[3]=points[ip].z - tz;
```

```
        /* omega */
        MM(m,mw,3,3,3,mm);
        Ab(mm,dx,3,3,v1);
        bdot[1][(lmn-1)*7 + 1]= -scale*v1[1];
        bdot[2][(lmn-1)*7 + 1]= -scale*v1[2];
        bdot[3][(lmn-1)*7 + 1]= -scale*v1[3];
        /* phi */
        MM(m,mp,3,3,3,mm);
        Ab(mm,dx,3,3,v1);
        bdot[1][(lmn-1)*7 + 2]= -scale*v1[1];
        bdot[2][(lmn-1)*7 + 2]= -scale*v1[2];
        bdot[3][(lmn-1)*7 + 2]= -scale*v1[3];
        /* kappa */
        MM(mk,m,3,3,3,mm);
        Ab(mm,dx,3,3,v1);
        bdot[1][(lmn-1)*7 + 3]= -scale*v1[1];
        bdot[2][(lmn-1)*7 + 3]= -scale*v1[2];
        bdot[3][(lmn-1)*7 + 3]= -scale*v1[3];
        /* tx */
        v2[1]=1.0;
        v2[2]=0.0;
        v2[3]=0.0;
        Ab(m,v2,3,3,v1);
        bdot[1][(lmn-1)*7 + 4]=scale*v1[1];
        bdot[2][(lmn-1)*7 + 4]=scale*v1[2];
        bdot[3][(lmn-1)*7 + 4]=scale*v1[3];
        /* ty */
        v2[1]=0.0;
        v2[2]=1.0;
        v2[3]=0.0;
        Ab(m,v2,3,3,v1);
        bdot[1][(lmn-1)*7 + 5]=scale*v1[1];
        bdot[2][(lmn-1)*7 + 5]=scale*v1[2];
        bdot[3][(lmn-1)*7 + 5]=scale*v1[3];
        /* tz */
        v2[1]=0.0;
        v2[2]=0.0;
        v2[3]=1.0;
        Ab(m,v2,3,3,v1);
        bdot[1][(lmn-1)*7 + 6]=scale*v1[1];
        bdot[2][(lmn-1)*7 + 6]=scale*v1[2];
        bdot[3][(lmn-1)*7 + 6]=scale*v1[3];
        /* scale */
        Ab(m,dx,3,3,v1);
        bdot[1][(lmn-1)*7 + 7]= -v1[1];
        bdot[2][(lmn-1)*7 + 7]= -v1[2];
        bdot[3][(lmn-1)*7 + 7]= -v1[3];
        /* x */
        bddot[1][1]= -bdot[1][(lmn-1)*7 + 4];
        bddot[2][1]= -bdot[2][(lmn-1)*7 + 4];
        bddot[3][1]= -bdot[3][(lmn-1)*7 + 4];
        /* y */
        bddot[1][2]= -bdot[1][(lmn-1)*7 + 5];
        bddot[2][2]= -bdot[2][(lmn-1)*7 + 5];
        bddot[3][2]= -bdot[3][(lmn-1)*7 + 5];
        /* z */
        bddot[1][3]= -bdot[1][(lmn-1)*7 + 6];
        bddot[2][3]= -bdot[2][(lmn-1)*7 + 6];
        bddot[3][3]= -bdot[3][(lmn-1)*7 + 6];
        /* f */
        Ab(m,dx,3,3,v1);
        f[1]= -(obs[io].xm - scale*v1[1]);
        f[2]= -(obs[io].ym - scale*v1[2]);
        f[3]= -(obs[io].zm - scale*v1[3]);
```

```
        /* form normals partition contributions */

        AtA(bdot,3,nmod*7,ndot_add);
        Atb(bdot,f,3,nmod*7,tdot_add);
        AtA(bddot,3,3,nddot_add);
        Atb(bddot,f,3,3,tddot_add);
        AtB(bdot,bddot,3,nmod*7,3,nbar_add);

        test=strcmp(points[ip].point_name,"L");
        if(test == 777)
          {
          Print_Matrix(bdot,3,nmod*7,"point L matrix bdot");
          Print_Matrix(bddot,3,3,"point L matrix bddot");
          Print_Matrix(ndot_add,nmod*7,nmod*7,"point L matrix ndot contribution");
          Print_Matrix(nddot_add,3,3,"point L matrix nddot");
          Print_Matrix(nbar_add,nmod*7,3,"point L matrix nbar");
          Print_Vector(f,3,"point L vector f");
          Print_Vector(tdot_add,nmod*7,"point L vector tdot contribution");
          Print_Vector(tddot_add,3,"point L vector tddot");
          }

        /* accumulate into the normals */

        for(i=1; i<=nmod*7; i++)
          {
          for(j=1; j<=nmod*7; j++)
            {
            ndot[i][j]=ndot[i][j] + ndot_add[i][j];
            }
          for(j=1; j<=3; j++)
            {
            nbar[ip][i][j]=nbar[ip][i][j] + nbar_add[i][j];
            }
          tdot[i]=tdot[i] + tdot_add[i];
          }

        for(i=1; i<=3; i++)
          {
          for(j=1; j<=3; j++)
            {
            nddot[ip][i][j]=nddot[ip][i][j] + nddot_add[i][j];
            }
          tddot[ip][i]=tddot[ip][i] + tddot_add[i];
          }

        } /* process an observation for this point */
      } /* io=... loop through all observations */

    /* zero rows and columns of nddot, nbar, tddot for a control point */

    for(k=1; k<=3; k++)
      {
      if(points[ip].control_flag[k] == 1)
        {
        for(i=1; i<=nmod*7; i++) nbar[ip][i][k]=0.0;
        for(i=1; i<=3; i++) nddot[ip][i][k]=0.0;
        for(i=1; i<=3; i++) nddot[ip][k][i]=0.0;
        nddot[ip][k][k]=1.0;
        tddot[ip][k]=0.0;
        }
      )

    /* make point reduction computations */

    Gauss_Inverse(nddot[ip],3);
```

```
    AQAt(nbar[ip],nddot[ip],nmod*7,3,ndot_add);
    MM(nbar[ip],nddot[ip],nmod*7,3,3,nbar_add);
    Ab(nbar_add,tddot[ip],nmod*7,3,tdot_add);
    test=strcmp(points[ip].point_name,"L");
    if(test == 777)
      {
      Print_Matrix(ndot_add,nmod*7,nmod*7,"point L matrix ndot reduction");
      Print_Vector(tdot_add,nmod*7,"point L vector tddot reduction");
      }

    for(i=1; i<=nmod*7; i++)
      {
      for(j=1; j<=nmod*7; j++)
        {
        ndot[i][j]=ndot[i][j] - ndot_add[i][j];
        }
      tdot[i]=tdot[i] - tdot_add[i];
      }

    } /* ip=... loop through points */

  /* solve ndot */

/*
  for(i=1; i<=nmod*7; i++)
    {
    printf("ndot ");
    for(j=1; j<=7; j++)
      {
      printf(" %10.3lf",ndot[i][j]);
      }
    printf("\n");
    }
  printf("\n");

  for(i=1; i<=nmod*7; i++)
    {
    printf("ndot ");
    for(j=8; j<=14; j++)
      {
      printf(" %10.3lf",ndot[i][j]);
      }
    printf("\n");
    }
  printf("\n");

  for(i=1; i<=nmod*7; i++)
    {
    printf("ndot ");
    for(j=15; j<=21; j++)
      {
      printf(" %10.3lf",ndot[i][j]);
      }
    printf(" %10.3lf",tdot[i]);
    printf("\n");
    }
  printf("\n");
*/

  Gauss(ndot,tdot,nmod*7,delta_dot);
/*  for(i=1; i<=nmod*7; i++)
    printf("del= %lf\n",delta_dot[i]);
  printf("\n");
*/
```

```
/*
  while(!kbhit());
  ch=getch();
*/

  /* update the model parameters */

  avg_count=0;
  avg_dx=0.0;
  avg_dy=0.0;
  avg_dz=0.0;
  for(i=1; i<=nmod; i++)
    {
    models[i].omega+= delta_dot[(i-1)*7 + 1];
    models[i].phi  += delta_dot[(i-1)*7 + 2];
    models[i].kappa+= delta_dot[(i-1)*7 + 3];
    models[i].tx   += delta_dot[(i-1)*7 + 4];
    models[i].ty   += delta_dot[(i-1)*7 + 5];
    models[i].tz   += delta_dot[(i-1)*7 + 6];
    models[i].scale+= delta_dot[(i-1)*7 + 7];
    avg_dx=avg_dx + delta_dot[(i-1)*7 + 4];
    avg_dy=avg_dy + delta_dot[(i-1)*7 + 5];
    avg_dz=avg_dz + delta_dot[(i-1)*7 + 6];
    avg_count++;
    }

  /* now update the object point parameters */

  for(ip=1; ip<=npnt; ip++)
    {
    Atb(nbar[ip],delta_dot,nmod*7,3,tddot_add);
    for(i=1; i<=3; i++) tddot_add[i]=tddot[ip][i] - tddot_add[i];
    Ab(nddot[ip],tddot_add,3,3,delta_ddot);
    points[ip].x=points[ip].x + delta_ddot[1];
    points[ip].y=points[ip].y + delta_ddot[2];
    points[ip].z=points[ip].z + delta_ddot[3];
    avg_dx=avg_dx + fabs(delta_ddot[1]);
    avg_dy=avg_dy + fabs(delta_ddot[2]);
    avg_dz=avg_dz + fabs(delta_ddot[3]);
    avg_count++;
/*
    printf("point deltas %s %lf %lf %lf\n",points[ip].point_name,
            delta_ddot[1],delta_ddot[2],delta_ddot[3]);
*/

/*
    while(!kbhit());
    ch=getch();
*/
    }
  avg_dx=avg_dx/avg_count;
  avg_dy=avg_dy/avg_count;
  avg_dz=avg_dz/avg_count;
  printf("deltas iteration %5d %14.6lf %14.6lf %14.6lf\n",iter,
          avg_dx,avg_dy,avg_dz);


  } /* iteration loop */

vec_free(v1);
mat_free(m,3);

for(ip=1; ip<=npnt; ip++)
  (
  printf("%16s %12.3lf %12.3lf %12.3lf\n",points[ip].point_name,points[ip].x,
```

```
        points[ip].y,points[ip].z);
  }

} /* end of main */

/* *********** */
/* subroutines */
/* *********** */

int get_logical_model_number(int mn, int allow_add)
{
int i,found,logical_number;

found=0; /* false */
for(i=1; i<=nmod; i++)
  {
  if(models[i].model_name == mn)
    {
    found=1; /* true */
    logical_number=i;
    }
  }
if(found == 0)
  {
  if (allow_add)
    {
    nmod++;
    models[nmod].model_name=mn;
    }
  logical_number= -nmod;
  }
return(logical_number);
}

int get_logical_point_number(char *s, int allow_add)
{
int i,found,logical_number;

found=0; /* false */
for(i=1; i<=npnt; i++)
  {
  if(strcmp(points[i].point_name,s) == 0)
    {
    found=1; /* true */
    logical_number=i;
    }
  }
if(found == 0)
  {
  if (allow_add)
    {
    npnt++;
    strcpy(points[npnt].point_name,s);
    }
  logical_number= -npnt;
  }
return(logical_number);
}
```

```
/*
  Bethel - 15-JAN-93

  gauss.c - Gauss equation solver and matrix inversion
*/

#include "mat.h"


void Triangular_Decomposition (double **A, int n, double **UL, int *IPS)
{
 double *Scales;
 int i,j,k;
 int kp,kp1, nm1;
 int ip, idxpiv;
 double row_norm, big, size, pivot, em;

 Scales = vec_alloc(n);

 for (i=1; i<=n; i++)
  {
   IPS[i] = i ;
   row_norm = 0.0 ;
   for (j=1; j<=n; j++)
    {
     UL[i][j] = A[i][j];
     if (fabs(UL[i][j]) > row_norm)
       row_norm = fabs(UL[i][j]);
    }
   if (row_norm == 0.0)
     {
      printf ("error: matrix with zero row in Triangular_Decomposition\n");
      Scales[i] = 0.0 ;
     }
   else Scales[i] = 1.0/row_norm ;
  }

 nm1 = n - 1 ;
 for (k=1; k<=nm1; k++)
  {
   big = 0.0 ;
   for (i=k; i<=n; i++)
    {
     ip = IPS[i] ;
     size = fabs(UL[ip][k])*Scales[ip] ;
     if (size > big)
      (
       big = size ;
       idxpiv = i ;
      )
    }
  if (big == 0.0)
    printf ("error: singular matrix in Triangular_Decomposition\n");
  else
   (
    if (idxpiv != k)
     {
      j = IPS[k] ;
      IPS[k] = IPS[idxpiv] ;
      IPS[idxpiv] = j ;
     }
    kp = IPS[k] ;
    pivot = UL[kp][k] ;
    kp1 = k + 1 ;
    for (i=kp1; i<=n; i++)
```

```
      {
       ip = IPS[i] ;
       em = -UL[ip][k]/pivot ;
       UL[ip][k] = -em ;
       for (j=kp1; j<=n; j++)
         UL[ip][j] += em*UL[kp][j] ;
      }
    }
  }

  kp = IPS[n] ;
  if (UL[kp][n] == 0.0)
    printf ("error: singular matrix in Triangular_Decomposition\n") ;

  vec_free (Scales);
}

void Solve (double **UL, double *B, int n, int *IPS, double *X)
{
  int i, j;
  int np1;
  int ip, ip1;
  int iback, im1;
  double sum;

  np1 = n + 1 ;
  ip = IPS[1] ;
  X[1] = B[ip] ;
  for (i=2; i<=n; i++)
   {
    ip = IPS[i] ;
    im1 = i - 1 ;
    sum = 0.0 ;
    for (j=1; j<=im1; j++)
      sum += UL[ip][j]*X[j] ;
    X[i] = B[ip] - sum ;
   }

  ip = IPS[n] ;
  X[n] = X[n]/UL[ip][n] ;

  for (iback=2; iback<=n; iback++)
   {
    i = np1 - iback ;
    ip = IPS[i] ;
    ip1 = i + 1 ;
    sum = 0.0 ;
    for (j=ip1; j<=n; j++)
      sum += UL[ip][j]*X[j] ;
    X[i] = (X[i] - sum)/UL[ip][i] ;
   }
}

void Gauss (double **A, double *B, int n, double *X)
{
  double **UL;
  int *IPS;
  int i,j;

  UL = mat_alloc (n,n);
  IPS = ivec_alloc (n);

  Triangular_Decomposition (A, n, UL, IPS) ;
  Solve (UL, B, n, IPS, X) ;
```

```
 ivec_free (IPS);
 mat_free  (UL,n);
}


void Gauss_Inverse (double **A, int n)
{
 double **UL;
 double *Baux;
 double *X;
 int *IPS;
 int i,j;

 UL = mat_alloc (n,n);
 IPS = ivec_alloc (n);
 Baux = vec_alloc (n);
 X = vec_alloc(n);

 Triangular_Decomposition (A, n, UL, IPS) ;

 for (i=1; i<=n; i++)
  {
   for (j=1; j<=n; j++)
     Baux[j] = 0.0;
   Baux[i] = 1.0;
   Solve (UL, Baux, n, IPS, X);
   for (j=1; j<=n; j++)
     A[j][i] = X[j];
  }

  vec_free (X);
  vec_free (Baux);
  ivec_free (IPS);
  mat_free (UL,n);

}
```

```
/*
   Bethel - 15-JAN-93

   mat.h - header file for "mat.c" and "gauss.c"
*/

#ifdef ANSI
#define VOID void
#else
#define VOID
#endif

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

double *vec_alloc (int n);
void vec_free (double *V);
int *ivec_alloc (int n);
void ivec_free (int *V);
double **mat_alloc (int m, int n);
void mat_free(double **A, int nrows);
char **cmat_alloc(int m, int n);
void cmat_free(char **A, int nrows);
void AtA(double **A, int m, int n, double **L);
void AtB(double **A, double **B, int m, int n, int l, double **L);
void Atb(double **A, double *b, int m, int n, double *L);
void Ab(double **A, double *b, int m, int n, double *L);
void AtWA(double **A,double **W,int m,int n, double **L);
void AtWB(double **A,double **W,double *B,int m,int n, double *R);
void AQAt(double **A,double **Q,int m, int n, double **Qe);
void MM(double **A,double **B, int m,int n,int l,double **AB);
void ROTM(double **M, double omega, double phi, double kappa);
void Gauss (double **A, double *B, int n, double *X);
void Triangular_Decomposition (double **A, int n, double **UL, int *IPS);
void Solve (double **UL, double *B, int n, int *IPS, double *X);
void Gauss_Inverse (double **A, int n);
void HouseholderReduction(double **a, int m, int n);
void Qv(double **a, double *b, int m, int n);
void TriSolve(double **a, double *b, double *x, int m, int n);
void h1(int p, int l, int m, double *v, double *h);
void h2(int p, int l, int m, double *v, double h, double *c);
void Print_Matrix(double **A, int m, int n, char *s);
void Print_Vector(double *b, int n, char *s);
```

```
/*
  Bethel - 15-JAN-93

  mat.c - Matrix Routines
*/

#ifdef ANSI
#define VOID void
#else
#define VOID
#endif

#include <stdlib.h>
#include "mat.h"


/*... Function vec_alloc ...*/
/* allocates space for a vector V of n double precision values */
/* returns the pointer to V, access elements V[1..n] */

double *vec_alloc (int n)
{
 double *V ;
 V = (double *) calloc(n, sizeof(double)) ;
 if (V==NULL)
  {
   fprintf (stderr, "could not allocate memory") ;
   exit (1) ;
  }
 return (V-1) ;
}

/*... Function vec_free ...*/
/* free the space used by a vector V of n double precision values */

void vec_free(double *V)
{
 free(V+1);
}

/*... Function ivec_alloc ...*/
/* allocates space for a vector IV of n integer values */
/* returns the pointer to IV, access elements IV[1..n] */

int *ivec_alloc (int n)
{
 int *IV ;
 IV = (int *) calloc(n, sizeof(int)) ;
 if (IV==NULL)
  {
   fprintf (stderr,"could not allocate memory") ;
   exit (1) ;
  }
 return (IV-1) ;
}

/*... Function ivec_free ...*/
/* free the space used by a vector IV of n integer values */

void ivec_free(int *IV)
{
 free(IV+1);
}

/*... Function mat_alloc ...*/
```

```
/* allocates space for a matrix A of mxn double precision values */
/* returns the pointer to A, access elements A[1..m][1..n] */

double **mat_alloc (int m, int n)
{
 int i ;
 double **A ;

 A = (double **) calloc (m, sizeof (double *)) ;
 if (A==NULL)
  (
   fprintf (stderr, "could not allocate memory") ;
   exit (1) ;
  }
 A -= 1 ;
 for (i=1; i<=m; i++)
  {
   A[i] = (double *) calloc (n, sizeof (double)) ;
   if (A[i]==NULL)
    {
     fprintf (stderr, "could not allocate memory") ;
     exit (1) ;
    }
   A[i] -= 1 ;
  }
 return A ;
}

/*... Function mat_free ...*/
/* free space used by a matrix A of mxn double precision values */
/* allocated by function mat_alloc */

void mat_free(double **A, int nrows)
(
 int i;

 for (i=1; i<=nrows; i++)
   free(A[i]+1);
 free(A+1);
}

/*... Function cmat_alloc ...*/
/* allocates space for a matrix A of mxn char values */
/* returns the pointer to A, access elements A[1..m][1..n] */

char **cmat_alloc (int m, int n)
{
 int i ;
 char **A ;

 A = (char **) calloc (m, sizeof (char *)) ;
 if (A==NULL)
  {
   fprintf (stderr, "could not allocate memory") ;
   exit (1) ;
  }
 A -= 1 ;
 for (i=1; i<=m; i++)
  {
   A[i] = (char *) calloc (n, sizeof (char)) ;
   if (A[i]==NULL)
    {
     fprintf (stderr, "could not allocate memory") ;
     exit (1) ;
    }
```

```
   A[i] -= 1 ;
  }
 return A ;
}

/*... Function cmat_free ...*/
/* free space used by a matrix A of mxn char values */
/* allocated by function mat_alloc */

void cmat_free(char **A, int nrows)
{
 int i;

 for (i=1; i<=nrows; i++)
   free(A[i]+1);
 free(A+1);
}

/* ... Function AtA ... */
   void AtA(double **A, int m, int n, double **L)
   {
   int i,j,k;
   double sum;

   for (i=1; i<=n; i++)
    {
     for (j=1; j<=n; j++)
      {
      sum = 0.0;
      for(k=1; k<=m; k++) sum=sum + A[k][i]*A[k][j];
      L[i][j]=sum;
      }
    }
   }

/* ... Function AtB ... */
   void AtB(double **A, double **B, int m, int n, int l, double **L)
   {
   int i,j,k;
   double sum;

   for (i=1; i<=n; i++)
    {
     for (j=1; j<=l; j++)
      {
      sum = 0.0;
      for(k=1; k<=m; k++) sum=sum + A[k][i]*B[k][j];
      L[i][j]=sum;
      }
    }
   }


/* ... Function Atb ... */
   void Atb(double **A, double *b, int m, int n, double *L)
   {
   int i,j,k;
   double sum;

   for (i=1; i<=n; i++)
     {
     sum = 0.0;
     for (k=1; k<=m; k++) sum=sum + A[k][i]*b[k];
     L[i]=sum;
     }
```

```
   }

/* ... Function Ab ... */
   void Ab(double **A, double *b, int m, int n, double *L)
   {
   int i,j,k;
   double sum;

   for (i=1; i<=m; i++)
     {
     sum = 0.0;
     for (k=1; k<=n; k++) sum=sum + A[i][k]*b[k];
     L[i]=sum;
     }
   }

/*... Function AtWA ...*/
/* computes the product of the matrices At.W.A, the LHS of normal equations */
/* A[1..m][1..n], W[1..m][1..m] */
/* returns the pointer to the result, L, access elements L[1..n][1..n] */

   void AtWA(double **A, double **W, int m, int n, double **L)
   {
   double *V;
   int i,j,k,l;

   V = vec_alloc (m) ;

   for (i=1; i<=n; i++)
    {
     for (k=1; k<=m; k++)
      {
       V[k] = 0.0;
       for (l=1; l<=m; l++)
         V[k] += A[l][i]*W[l][k];
      }
     for (j=1; j<=n; j++)
      {
        L[i][j] = 0.0;
        for ( k=1; k<=m; k++)
         L[i][j] += V[k]*A[k][j];
      }
    }
   vec_free(V);
   }

/*... Function AtWB ...*/
/* computes the product of the matrices At.W.B, the RHS of normal equations */
/* A[1..m][1..n], W[1..m][1..m], B[1..m] */
/* returns the pointer to the result, R, access elements R[1..n] */

   void AtWB(double **A, double **W, double *B, int m, int n, double *R)
   {
   double *V;
   int i,j,k,l;

   V = vec_alloc (m) ;

   for (i=1; i<=n; i++)
    {
     R[i] = 0.0;
     for (k=1; k<=m; k++)
      {
       V[k] = 0.0;
       for (l=1; l<=m; l++)
```

```
          V[k] += A[l][i]*W[l][k];
        }
       for ( k=1; k<=m; k++)
        R[i] += V[k]*B[k];
      }
     vec_free(V);
     }


/*... Function AQAt ...*/
/* computes the product of the matrices A.Q.At, A[1..m][1..n], Q[1..n][1..n] */
/* returns the pointer to the result, Qe, access elements Qe[1..m][1..m] */

    void AQAt(double **A, double **Q, int m, int n, double **Qe)
    {
     int i, j, k, l;
     double *V;

     V = vec_alloc (n);

     for (i=1; i<=m; i++)
      {
       for (k=1; k<=n; k++)
        {
         V[k] = 0.0;
         for (l=1; l<=n; l++)
           V[k] += A[i][l]*Q[l][k];
        }
       for (j=1; j<=m; j++)
        {
         Qe[i][j] = 0.0;
         for (l=1; l<=n; l++)
           Qe[i][j] += V[l]*A[j][l];
        }
      }
     vec_free(V);
    }


/*... Function MM ...*/
/* multiplies two matrices A[1..m][1..n] B[1..n][1..l] */
/* returns the pointer to AB, access elements AB[1..m][1..l] */

    void MM(double **A, double **B, int m, int n, int l, double **AB)
   {
    int i,j,k;

    for (i=1; i<=m; i++)
      for (j=1; j<=l; j++)
          AB[i][j] = 0.0;
    for (i=1; i<=m; i++)
      for (j=1; j<=n; j++)
        for (k=1; k<=l; k++)
          AB[i][k] += A[i][j] * B[j][k] ;

    }


/*... Function ROTM ...*/
/* computes the omega, phi, kappa rotation matrix, M */
/* returns the pointer to M, access elements M[1..3][1..3] */

    void ROTM(double **M, double omega, double phi, double kappa)
    {
    double sw, sp, sk;
```

```
    double cw, cp, ck;

    sw = sin (omega);
    cw = cos (omega);
    sp = sin (phi);
    cp = cos (phi);
    sk = sin (kappa);
    ck = cos (kappa);

    M[1][1] = cp*ck;
    M[2][1] = -cp*sk;
    M[3][1] = sp;
    M[1][2] = cw*sk + sw*sp*ck;
    M[2][2] = cw*ck - sw*sp*sk;
    M[3][2] = -sw*cp;
    M[1][3] = sw*sk - cw*sp*ck;
    M[2][3] = sw*ck + cw*sp*sk;
    M[3][3] = cw*cp;

    }

/*... Function Print_Matrix ...*/

void Print_Matrix(double **A, int m, int n, char *s)
{
int i,j,k;
int cc;
div_t qr;

printf("%s\n\n",s);
qr=div(n,7);
cc=1;
for(k=1; k<=qr.quot; k++)
  {
  for(i=1; i<=m; i++)
    {
    for(j=cc; j<=cc+6; j++) printf(" %10.4lf",A[i][j]);
    printf("\n");
    }
  printf("\n");
  cc=cc + 7;
  }

if(qr.rem != 0)
  {
  for(i=1; i<=m; i++)
    {
    for(j=cc; j<=n; j++) printf(" %10.4lf",A[i][j]);
    printf("\n");
    }
  printf("\n");
  }
}

void Print_Vector(double *b, int n, char *s)
{
int i;

printf("%s\n\n",s);
for(i=1; i<=n; i++)
  {
  printf(" %10.4lf\n",b[i]);
  }
printf("\n");
}
```

COVER DESIGN BY ALDO GIORGINI